The Dissertation Committee for Gökberk Kabacaoğlu
certifies that this is the approved version of the following dissertation:

Numerical Methods for Simulations and Optimization of Vesicle Flows in Microfluidic Devices

Committee:

George Biros, Supervisor

Omar Ghattas

Robert Moser

Michael Shelley

Numerical Methods for Simulations and Optimization of Vesicle Flows in Microfluidic Devices

by

Gökberk Kabacaoğlu

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2019

To my mother, Gülsevim Kabacaoğlu.

Acknowledgments

I want to thank my advisor Prof. George Biros for his extensive support and mentorship. The years of working with him, his vision, insight, and knowledge brought me valuable experiences. I am very grateful for the opportunities he generously provided. Additionally, I would like to thank Prof. Bryan Quaife who was a postdoctoral researcher in the PADAS group, when I joined the group. He helped me discover my new research area quickly. He was always available and helpful.

I would like to thank my great teachers and mentors back home in Turkey. They have taught the importance of being disciplined, organized and self-confident. My experiences as an undergraduate at Bilkent University were life-changing. Among my professors at Bilkent, I want to thank Prof. İlker Temizer, particularly. He has done amazing things for me, which eventually led me to pursue a doctoral degree.

I want to thank to my committee members. It is a great honor to have their support on my dissertation. I consider myself privileged for getting to know them. Many thanks to my labmates, in particular, Prof. Andreas Mang, James Levitt and Yuanxun Bao. I also want to thank to all the faculty and students at the Oden Institute for providing such an encouraging environment to work hard and learn more.

I have been incredibly fortunate to have amazing people in my life who love me, care about me and always give me their boundless support. My main motivation in life has been to make them happy and proud of me. I hope I will always have them

in my life. I sincerely thank and dedicate this thesis to them, and especially to my mother. She has been very strong, supportive and encouraging. I will always be deeply indebted to her.

Numerical Methods for Simulations and Optimization of Vesicle Flows in Microfluidic Devices

Gökberk Kabacaoğlu, Ph.D.
The University of Texas at Austin, 2019

Supervisor: George Biros

Vesicles are highly deformable particles that are filled with a Newtonian fluid. They resemble biological cells without a nucleus such as red blood cells (RBCs). Vesicle flow simulations can be used to design microfluidic devices for medical diagnoses and drug delivery systems. This dissertation focuses on efficient numerical methods for simulations and optimization of vesicle flows in two dimensions.

We consider flows with very low Reynolds numbers and inextensible vesicle membranes that resist bending. Our numerical scheme is based on a boundary integral formulation which is known to be efficient for such flows. This formulation leads to a set of nonlinear integro-differential equations for the vesicle dynamics. Complex interplay between the nonlocal hydrodynamic forces and the membranes' elasticity determines the vesicles' motion. Many state-of-the-art numerical schemes can resolve these complex flows. However, simulations remain computationally expensive since high-resolution discretization is needed. The high computational cost limits the use of the simulations for practical purposes such as optimization.

Our first attempt to reduce the cost is to use low-resolution discretization. We present a scheme that systematically integrates several correction algorithms that are necessary for stable and accurate low-resolution simulations. We compare the low-resolution simulations with their high-fidelity counterparts. We observe that our scheme enables both fast and statistically accurate simulations.

We accelerate vesicle flow simulations further by replacing expensive parts of the numerical scheme with low-cost function approximations. We propose a machine-learning-augmented reduced model that uses several multilayer perceptrons to model different aspects of the flows. Although we train the perceptrons with high-fidelity single-particle simulations for one time step, our method enables us to conduct long-horizon simulations of suspensions with several particles in confined geometries. It is faster than a state-of-the-art numerical scheme having the same number of degrees of freedom and can reproduce several features of the flow accurately. It generalizes as is to other particles like deformable capsules, drops, filaments and rigid bodies.

Moreover, we investigate deformability-based sorting of RBCs using a microfluidic device that enables medical diagnoses of diseases such as malaria. Using our numerical scheme we solve a design optimization problem to find optimal designs of the device that provide efficient sorting of cells with arbitrary mechanical properties.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction / Executive Summary

We study numerical methods for simulations of Stokesian flows with deformable particles suspended in incompressible Newtonian fluids. Such flows are ubiquitous in biology, e.g., flows of drops, capsules, cells, slender bodies, filaments, active swimmers. We consider a specific particle type, a vesicle. Simulation of vesicles is essential for many industrial and biological applications, e.g., understanding the microcirculation of blood, thrombosis risk assessment and optimal design of microfluidic devices. Analytical solutions to the boundary value problem of vesicle flows exist only for a few simple cases. That is why numerical simulations and experiments are the only tools for their quantitative study. Many existing state-of-the-art numerical schemes can resolve complex vesicle flows. However, even when using provably optimal algorithms, these simulations can be computationally expensive, especially for suspensions with a large number of vesicles. The high computational cost limits the use of simulations for parameter exploration, optimization, or uncertainty quantification. Here we present two fast numerical schemes for Stokesian particulate flows in two dimensions. One of them systematically integrates several algorithmic fixes to improve stability and accuracy of simulations using low-resolution discretization in space and time. The proposed scheme is stable and accurately captures underlying physics without high resolutions. Thereby, it provides 10×-100× speed-up. The other one is a machine-

learning-augmented reduced order model. This scheme accelerates the simulations even further by replacing the most computationally expensive parts of the numerical scheme with low-cost function approximations. It provides stable and accurate simulations in case where the low-resolution scheme produces unacceptable errors since the machine learning model is trained using high-resolution simulations. Using our first scheme, we also investigate (i) mixing in vesicle suspensions, (ii) sorting particles by their mechanical properties using a microfluidic device and (iii) optimal design of the device for efficient sorting of particles with arbitrary mechanical properties.

1.1 Background

Vesicles are micron-sized particles encapsulating and suspended in incompressible fluids. Their simulation plays an important role in many biological applications [94, 154], such as biomembranes [153] and red blood cells (RBCs) [50, 83, 118, 124, 136]. Vesicle membrane is a lipid bilayer with a thickness of approximately 5 nm, which is very small compared to the vesicle radius ($\approx 10\,\mu$m). The bilayer is highly deformable as it is in a liquid phase at room temperature [35]. The membrane resists bending and stretching. It is considered locally inextensible as its stretching stiffness is large [155]. Additionally, the membrane is impermeable to many molecules except water. However, if a net flow exists, volume changes have a longer time scale compared to the typical experimental time scale of about 15 minutes. Therefore, a vesicle's volume is almost constant [63]. In equilibrium vesicles show wide variety of shapes resulting from the minimization of the bending energy [1, 63]

$$\mathscr{H} = \int_A \left(\frac{\kappa_b}{2}(2H - C_0)^2 + \kappa_G K \right) dA,$$

where H and K are the mean and Gaussian curvatures, respectively, κ_b and κ_G the corresponding bending stiffnesses, C_0 a spontaneous curvature, A the membrane surface area. In external flow field vesicles show rich and complex dynamics that depends on an interplay between membrane elasticity and hydrodynamic forces. While vesicles naturally exist in biological systems, they can also be artificially manufactured [17, 38]. Although biological cells have more complex structures, vesicles have been used as a model system for anucleate cells such as red blood cells (RBCs) [119, 124].

1.2 Numerical Model

We consider two-dimensional (2D) vesicle flows with small velocity and length scales, i.e., the Reynolds number $Re \ll 1$, so that the inertial effects may be neglected. Thereby, we model vesicle flows using a quasi-static Stokes approximation scheme [31, 94]. We assume that the suspending and the encapsulated fluids are Newtonian. We use a standard mechanical model for vesicles, i.e., the locally inextensible vesicle membrane resists bending and stretching, and is impermeable to the suspending fluid. These assumptions are acceptable to study real phenomena involving blood flows in small capillaries where the particulate nature of blood is important as the diameter of the capillary ($\approx 80\,\mu$m) is in the same order of the diameter of RBCs ($\approx 8\,\mu$m) [52]. We use a boundary integral formulation for vesicle flows. Although many other methods can be used [47, 104], boundary integral methods are known to work very well for Stokesian fluids with deformable interfaces [45, 47, 140, 141, 182]. In this formulation, only the vesicle membranes and fixed boundaries are discretized and the trajectories of Lagrangian points on the vesicle membranes are tracked. The evolution of the i^{th}

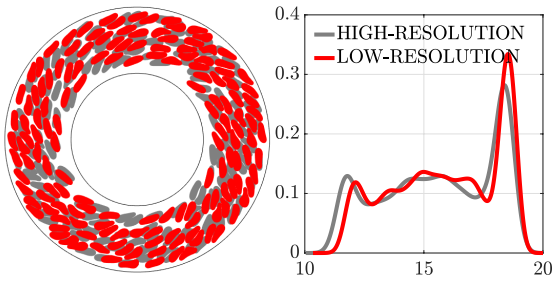
membrane, γ_i, is governed by

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{u}_{\text{self}}[\gamma_i](\mathbf{x}) + V[\gamma_i]\mathbf{u}_\infty[\gamma, \Gamma](\mathbf{x}), \tag{1.1}$$

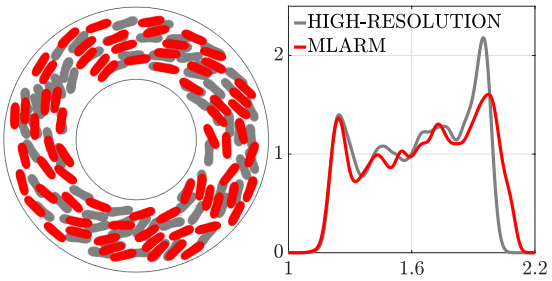
where $\mathbf{u}_{\text{self}}$ is the velocity induced by the i^{th} vesicle itself (e.g., relaxation of a single vesicle in a stationary flow involves only this term) and $\mathbf{u}_\infty[\gamma, \Gamma]$ is either the imposed flow analytically given in the case of unbounded flows or the velocity induced by other vesicle membranes γ and the fixed boundaries Γ. $V[\gamma_i]\mathbf{u}_\infty[\gamma, \Gamma]$ evaluates the effect of $\mathbf{u}_\infty$ on γ_i. Brackets denote that computing the preceding term requires solving integral equations involving boundaries in the brackets. Our group has already developed novel algorithms based on the boundary integral formulation for 2D vesicle flows in bounded and unbounded domains [142, 143, 146, 164] and the 3D flows in unbounded domains [111, 165].

1.3 Contributions

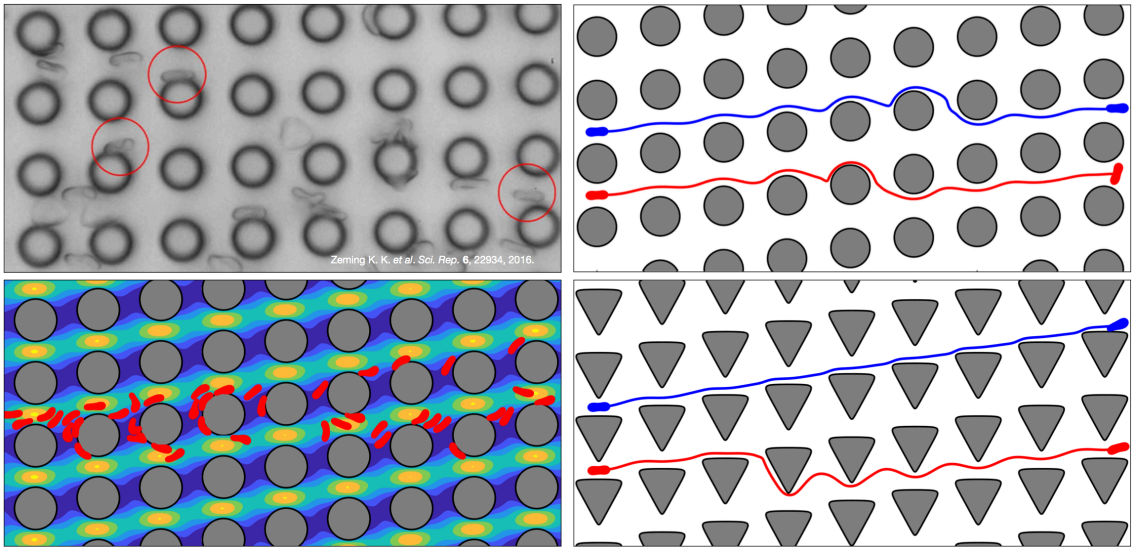
The author has contributed to the field of numerical methods for vesicle flows and the study of physical phenomena related to such flows in the form of published articles that are summarized below (also see Fig. 1.1 for visualizations from these articles).



(a) Low-resolution simulations



(b) Machine-learning-augmented simulations



(c) Simulation and design of a microfluidic device for cell sorting

Figure 1.1: *(a) Dense vesicle suspension in a Taylor-Couette flow simulated using our fast numerical scheme based on low-resolution spatio-temporal discretization. We superimpose low-resolution simulation (red) and its high-resolution counterpart (gray). The low-resolution simulation accurately captures the flow statistics, i.e., probability distribution of vesicles' centers (the right figure). (b) Dense vesicle flows simulated using our machine-learning-augmented reduced model. We accelerate vesicle flow simulations even further using machine learning algorithms. (c) Simulation and design of a deterministic lateral displacement device for cell sorting. The device identifies diseased RBCs using the fact that they are stiffer than the healthy ones. Top left shows an actual experimental DLD device. Bottom left one is our simulation. Top right shows a standard device (with cylindrical pillars) that cannot sort cells. Bottom right shows an optimal design that achieves sorting.*

Low-resolution simulations of vesicle suspensions in 2D [80]. One way to reduce the computational cost of vesicle flow simulations is to use low-resolution spatio-temporal discretization. However, simply reducing the resolution results in unstable simulations and nonphysical results. In this paper, we focus on algorithms that are necessary to maintain stability of simulations, all the while accurately capturing the statistics of the underlying flow using as coarse discretization as possible. The main contributions are (i) developing a robust scheme implementing some standard techniques and introducing new algorithms, (ii) heuristically calibrating the parameters of these algorithms to develop a black-box solver that can capture underlying physics accurately, and (iii) systematic error analysis of low-resolution simulations. The proposed scheme delivers statistically accurate low-resolution simulations of dense suspensions (see Fig. 1.1(a)), while it can be 10× to 100× faster.

One of the correction algorithms we introduce is to determine the upsampling rate that is sufficient to avoid aliasing errors. Additionally, we reformulate the membrane reparameterization algorithm used in 3D [165] for 2D simulations. The algorithm is essential to maintain a grid quality. Another one is an adaptive time stepping algorithm that can be used at all resolutions. This is a variation of our group's previous work [144] which relies on assumptions that are not valid at the low resolutions. A vesicle's area and arc-length are invariant in 2D. However, numerical errors at low resolutions accumulate and result in large changes in them. We present an efficient algorithm to correct these errors without modifying the governing equations. Near-field (lubrication like) hydrodynamic interactions, if accurately captured, avoid vesicles' collision. However, they cannot be resolved accurately at low resolutions. We

detect collisions with spectral accuracy [142] and implement a short range repulsion force to penalize the close proximity of vesicles to each other.

Quantification of mixing in vesicle suspensions using numerical simulations in two dimensions [79]. We study mixing in vesicle suspensions in a Taylor-Couette flow. We use the computational infrastructures described in the previous paragraph. Vesicles play an important role in intracellular and intercellular transport. Artificial vesicles are also used in drug-delivery systems. There has been very little work in characterizing mixing in vesicle flows. To the best of our knowledge, this study is one of the first studies investigating the effects of vesicle suspensions on mixing.

We simulate mixing of a passive scalar in flows with and without vesicles. We use a pseudo-spectral scheme for the passive advection-diffusion equation and measure mixing with negative Sobolev norms of the concentration. We study the effects of area fraction of vesicles, viscosity contrast between the fluids in the interior and the exterior of vesicles, initial condition of the solute, the Peclet number and mixing metric. We compare mixing in the Taylor-Couette flows with and without vesicles. On the one hand, the presence of vesicles in most cases slightly suppresses mixing. This is because the solute can be only diffused across the vesicle interface and not advected. On the other hand, there exist spatial distributions of the solute for which the unperturbed Taylor-Couette flow completely fails to mix by advection whereas the presence of vesicles provides strong advection. We derive a simple condition that relates the velocity and the initial condition of a solute and can be used to characterize the cases in which the presence of vesicles promotes mixing.

Sorting same-size red blood cells in deep deterministic lateral displacement devices [77]. Microfluidic sorting of deformable particles by their deformability finds many applications, for example, medical devices for cells. Deterministic lateral displacement (DLD) is one of them (see Fig. 1.1(c)) [71]. Size-based sorting of rigid spherical particles using DLD has been well studied experimentally and numerically [33, 34, 72]. However, the sorting of non-spherical deformable particles such as RBCs is more complicated than that of rigid particles. For example, is it possible to separate deformable particles that have the same size but different mechanical properties? To the best of our knowledge, this is one of the first studies investigating the effects of the complex RBC dynamics in DLD devices.

We present a high-fidelity DLD model that captures true physics without any tuning parameters. Using the model, we quantitatively characterize the physical mechanisms that enable the cell sorting and investigate the effects of the interior fluid viscosity and membrane elasticity of a cell on its behavior. We consider deep devices in which a cell can show rich dynamics such as taking a particular angular orientation depending on its mechanical properties. We have found that cells moving with a sufficiently high positive inclination angle with respect to the flow direction displace laterally while those with smaller angles travel with the flow streamlines. Thereby, deformability-based cell sorting is possible. The underlying mechanism here is cell migration due to the cell's positive inclination and curved flow lines. We also assess the efficiency of the technique for dense suspensions. It turns out that most of the cells in dense suspensions do not displace in the lateral direction no matter what their deformability is. As a result, sorting cells using a DLD device becomes harder.

Optimal design of deterministic lateral displacement device for viscosity-contrast-based cell sorting [76]. Given cells with arbitrary mechanical properties and constraints on device size, we find an optimal DLD design that can efficiently sort the cells by their deformability, in particular their viscosity contrast values. The main contribution of our study is to pose designing a DLD device as an optimization problem for discovering optimal designs systematically and to develop methods to solve the problem efficiently (see Fig. 1.1(c)).

The design parameters of a DLD device are the pillar cross section (i.e., top view of pillars), the tilt angle of pillar rows, and the center-to-center distances. These define a unique device. We fix the tilt angle and the center-to-center distances. So the only design parameter is the pillar cross section, which we parametrize with uniform fifth-order B-splines. We propose an objective function that assesses whether a design provides efficient cell sorting. We solve the problem using a stochastic optimization algorithm. Evaluating the objective function requires simulating cell flows through a DLD device. That is why it is infeasible to solve the optimization problem using our high-fidelity DLD model introduced in our previous study. So we propose a low-fidelity DLD model that enables fast solution of the problem. We consider four sorting scenarios involving cells with similar viscosity-contrast values. The optimal designs in these scenarios are different from those in the literature.

Machine learning acceleration of simulations of Stokesian suspensions [78]. In this paper, we continue our efforts to accelerate simulations of Stokesian particulate flows. The main contribution is a generic machine-learning-augmented reduced model

that replaces expensive parts of a numerical scheme with low-cost function approximations. Machine learning techniques provide useful tool for approximating nonlinear functions and among successful machine learning algorithms we opt for multilayer perceptrons (MLPs). Our model combines the MLPs trained with high-fidelity single-particle simulations for one time step and low-fidelity simulations. Although we train using single-particle, short-horizon, unconfined flows, our method enables us to conduct long-horizon simulations of suspensions with several particles in confined geometries (see Fig. 1.1(b)). We illustrate the performance of our model on integral equation formulation of vesicle suspensions in two dimensions. It is approximately an order of magnitude faster than a state-of-the-art numerical scheme having the same number of degrees of freedom and can reproduce several features of the flow quite accurately.

1.4 Limitations and Future Work

Main limitation of our studies is that they are in two dimensions. While our correction algorithms for low-resolution simulations can be extended to three dimensions [111], 3D simulations are still computationally expensive to investigate real physics problems, let alone optimization. We expect that our machine-learning-augmented reduced model can significantly speed up 3D simulations of Stokesian particulate flows. That is why, we plan to extend the reduced model to three dimensions as a future work.

Using dense suspensions in DLD devices would provide high-throughput sorting. However, the DLD technique fails to sort cells in dense suspensions. That is why, it is important to investigate optimal DLD designs that can be used to sort dense suspen-

sions. Simulations of dense suspensions in DLD are computationally expensive and we cannot use our low-fidelity DLD model for such simulations. Another future work is to use our machine-learning-augmented reduced model for such simulations and solve the optimization problem to find designs that can still sort dense suspensions.

1.5 Organization of the Thesis

We present each of the aforementioned publications separately in a chapter of this thesis. In Chapter 2, we describe our numerical scheme integrating the correction algorithms for stable and predictive low-resolution simulations of vesicles. We review the literature on numerical simulations of vesicle flows, present the governing equations, the integral equation formulation and our numerical scheme for the flows. In Chapter 3, we discuss quantification of mixing in vesicle suspensions and effects of vesicle flows on mixing. We also present a pseudo-spectral scheme for the advection-diffusion equation. In Chapter 4, we investigate deformability-based sorting of the same-size red blood cells in deterministic lateral displacement devices. In Chapter 5, we discuss optimal designs of these devices for efficient sorting of cells with arbitrary mechanical properties. In Chapter 6, we present the machine-learning-augmented reduced model for Stokesian suspensions and discuss its accuracy and speed in comparison to our low-resolution scheme introduced in Chapter 2.

Nomenclature. We define the symbols and notations at the beginning of each chapter depending on the context of a chapter.

Chapter 2

Low-Resolution Simulations of Vesicle Suspensions in Two Dimensions

In this chapter[1] we introduce equations governing vesicle flows in two dimensions (2D) and the corresponding boundary integral equation formulation. We, then, present a robust framework based on low-resolution discretization in space and time for fast simulations of vesicle flows. This low-resolution scheme is an extension of [142, 144]. Our low-resolution correction algorithms (LRCA) include anti-aliasing and membrane reparameterization for avoiding spurious oscillations in vesicles' membranes, adaptive time stepping and a repulsion force for handling vesicle collisions and, correction of vesicles' area and arc-length for maintaining physical vesicle shapes. We perform a systematic error analysis by comparing the low-resolution simulations of dilute and dense suspensions with their high-fidelity, fully resolved counterparts. We observe that the LRCA enables both efficient and statistically accurate low-resolution simulations of vesicle suspensions, while it can be 10× to 100× faster.

[1] This chapter is based on work that has been published in [80]. The authors equally contributed.

2.1 Introduction

Vesicle flows are characterized by large deformations, local inextensibility of a vesicle's membrane, conservation of enclosed area due to the incompressibility of the fluid inside the vesicle, and stiffness related to tension and bending forces. These features make simulating suspensions at low resolutions a challenging problem. Although many algorithmically optimal methods for vesicle flows exist (see below), the computational costs remain prohibitively expensive for large vesicle suspensions. So, the basic question we try to address is the following. What is the minimum resolution required to recover different quantities of interest in the context of boundary integral equation methods for vesicle suspensions? Understanding and improving low-resolution simulations will enable parametric studies and optimization (e.g., phase diagrams and design of microfluidic devices). Also many boundary integral equation codes use the empirical corrections we investigate here because convergence studies and high-resolution simulations are not possible. Further understanding these corrections and reducing the number of simulation parameters are valuable for the community.

We opt to study two-dimensional vesicle suspensions since convergence studies in three dimensions for suspensions with a large number of vesicles can be extremely expensive [111, 147]. In addition, two dimensional simulations are valuable on their own since they can reproduce experimentally observed flow physics in many regimes (e.g., motion of red blood cells in microchannels [40, 83], margination of white blood cells in blood flow [40, 46], and sorting of rigid particles and RBCs using deterministic lateral displacement technique [145, 175, 179]). Our previous results for simulating high-concentration vesicle suspensions in 2D focus on accurate quadrature and

high-order semi-implicit time stepping [142, 143]. The results in those papers rely on sufficient resolution and provide a robust framework for simulations. For example, vesicles do not collide because all hydrodynamic interactions are resolved with spectral accuracy. Thus, there is no need to introduce artificial repulsion forces between vesicles. We can accurately resolve long time horizon simulations for concentrated suspensions with roughly 96 or 128 points per vesicle. But in three dimensions such a resolution is prohibitively expensive. For example, a similar resolution using the 3D version of these algorithms would require over 10,000 points per vesicle [111]. Therefore, there is a need to use some empirical fixes to maintain stability in simulations, all the while accurately capturing the statistics of the underlying flow using as coarse discretization as possible. To measure the accuracy of the physics and statistics, we develop the algorithms in 2D so that we can compare with "ground truth" simulations performed at an adequate resolution. Demonstrating the effectiveness of these algorithms at low resolutions is the first step towards extending them to 3D.

Contributions. Low-resolution simulations of vesicle suspensions can become unstable as a result of spurious oscillations in vesicles' shapes due to computing nonlinear terms, nonphysical changes in vesicles' areas and arc-lengths, and vesicle collisions. We address these issues and develop a robust method by implementing some standard algorithms and also introducing new algorithms. We calibrate the parameters for these algorithms heuristically. We, then, investigate accuracy of our low-resolution simulations compared to the ground truth solutions. We also report the self-convergence of the low-resolution simulations. The numerical experiments help us develop a black-box

solver that can capture underlying physics accurately using as coarse discretization as possible without having to adjust parameters other than the spatial and temporal resolution. We summarize these contributions as follows:

- We introduce an efficient algorithm for determining an upsampling rate that is sufficient for controlling the aliasing errors caused by nonlinear terms, but not too large so that the computational costs are not unnecessarily inflated. Additionally, we formulate the reparameterization algorithm in [165] into 2D to avoid spurious oscillations in vesicle shapes.

- A vesicle's area and arc-length are invariant in two-dimensional vesicle simulations (their counterparts are volume and surface area in 3D). However, at low resolutions the errors can be extensive and hence result in unstable and nonphysical flows in time scales much shorter than the target time horizons. Therefore, we present an efficient scheme to correct those errors without modifying the governing equations.

- The adaptive time stepping work in [144], which uses errors in vesicles' area and arc-length in order to adjust the time step size, relied on asymptotic assumptions of the truncation error, which are not valid at the low resolutions. Since this result breaks down, we present a new variation of this scheme that can be used at all resolutions.

- Near-field (lubrication like) hydrodynamic interactions cannot be resolved accurately at low resolutions. This leads to nonphysical collisions between vesicles.

We detect collisions with spectral accuracy [142] and implement a short range repulsion force [62, 169] to keep vesicles sufficiently separated. Unlike many other repulsion models requiring two parameters, our scheme is parameter-free, i.e., the repulsion length scale is set beforehand based on numerical experiments and the strength of the force is adaptive.

- We calibrate all the parameters of the LRCA heuristically and thereby develop a black-box solver with a single parameter. The main parameters are the spatial resolution N, the temporal resolution, which is adjusted by the tolerance for the area-length errors ρ_{AL}, and a time budget T_{comp} so the solver can automatically set the minimum time steps. We test the solver in a real-world application of a microfluidic cell sorting device.

The summary of our observations from the numerical experiments is as follows: (i) All empirical fixes (anti-aliasing, reparameterization, repulsion, adaptive time stepping, area-length correction) are necessary to stabilize low-resolution simulations. Dropping one can result in failure. (ii) We do not have a way to guarantee convergence. Goal-oriented error estimation requires adjoints and we do not have this capability. The only way to check for convergence is to start with a coarse N and ρ_{AL} and refine until the results do not change significantly. Notice that this is also true for the fine-resolution simulations. Notice even in this scenario in which we compare simulations at different resolutions, the error metric matters a lot. If we are interested in convergence of individual trajectories, very refined simulations are necessary, especially for dense suspensions. But for error metrics that look at average quantities, (e.g., effective viscosity) convergence is faster and less sensitive to the details of the simulation.

Limitations. One limitation is that our results are entirely empirical. In general, there is a limited number of work on theoretical results for vesicles. Indeed the only results are for vesicles that are small perturbations of a disc and thus resemble rigid spheres. Although the algorithms are implemented in 2D, they can be easily extended to 3D: e.g., local area and length correction can be extended to a volume and surface area correction [111], and a surface reparameterization has already been implemented in 3D [111, 147, 165]. Another limitation is that our scheme is not directly applicable to inertial flows or suspensions in which the bulk fluid is non-Newtonian.

Related work. There is extensive work on vesicle simulations. We review those that are most relative to ours (see [47, 104] for several other methods). Youngren and Acrivos [176] introduced boundary integral equation methods for Stokes flow past a rigid particle. These methods are known to work very well for Stokesian fluids with deformable interfaces [45, 47, 140, 141, 182]. Pozrikidis [139] presents an extensive review of these methods for various Stokesian particulate flows. Our work relies on a formulation derived by Rallison and Acrivos [148] for two fluids separated by an interface and an integral equation formulation of the Stokes problem for a multiply-connected domain with Dirichlet boundary conditions derived by Power et al. [134, 135]. Our work is based on the line of work on 2D flows by our group. Let us summarize them. Veerapaneni et al. [164] considered vesicles in free-space flows with no viscosity contrast and presented a semi-implicit time stepping scheme that treats only intra-vesicle interactions implicitly. Rahimian et al. [146] extended this work to bounded flows with viscosity contrast. Then, Quaife and Biros [142] introduced

a robust framework to enable high-fidelity simulations of dense suspensions. They proposed a time stepping scheme that implicitly couples vesicle-vesicle and vesicle-fixed boundary interactions. They also introduced a fifth-order accurate integration scheme with optimal work for nearly-singular integrals for close vesicle-vesicle and vesicle-fixed boundary interactions. Finally, Quaife and Biros [144] presented an adaptive high-order accurate time stepping scheme based on spectral deferred correction method. They used invariant properties of vesicles, constant area and arc-length, to reduce the computational cost of error estimation for adaptive time stepping. The review of the literature on anti-aliasing techniques, surface reparameterization algorithms, area-length correction methods, repulsion models, and error measures for vesicle dynamics and rheology is below.

Classical works in *anti-aliasing* (or de-aliasing) include [25, 95, 131]. In [126] and [127], if the discretization is with N points, the nonlinear terms are computed at a higher resolution ($1.5N$ points) and filtered back to N points. While this removes aliasing errors due to quadratic operations, the nonlinearities in the vesicle model, such as roots and inverses, are much stronger. Therefore, it is essential to find required upsampling rates. In [147], an algorithm that automatically adjusts the upsampling rate for differentiation is based on the mean curvature of the three-dimensional vesicles; our upsampling scheme is similar. It efficiently determines the sufficient upsampling rate for each vesicle to compute the force due to bending while we always upsample to $N^{3/2}$ to compute the layer potentials.

Using *membrane reparameterization* preserves grid quality and helps control aliasing errors. An algorithm for distributing grid points equally in arc-length for 2D

membranes is presented in [14] and implemented in [70, 112]. Additionally, Rahimian et al. [147] presents a reparameterization scheme for three-dimensional vesicles which redistributes points so that high-frequency components of the spectral discretization are minimized. Our reparameterization scheme is based on that work and smooths vesicle shapes by penalizing its high frequencies. We have observed that this provides better grid quality than equally spacing the points in arc-length.

Despite the local inextensibility and incompressibility conditions, *errors in the area and arc-length of a vesicle* can become large because of error accumulating at each time step. This not only results in nonphysical vesicle shapes, but can also lead to instabilities. In [112], this issue is addressed by performing area-length correction by modifying the governing equations. The arc-length is corrected by adding a correction term to the inextensibility condition and the area correction requires solving a quadratic equation. In [4, 18, 21], area and arc-length errors are corrected by adding artificial forces. Unlike those techniques, our area-length correction scheme does not modify the governing equations. We correct these errors after every time step by solving a constrained optimization problem. This scheme is also extended to three-dimensions in [111].

There are two common approaches for *handling collision* in granular media and particulate flows. Optimization-based contact resolution methods are frequently used in multibody contact mechanics for their robustness and efficiency. Lu et al. [109] have recently applied such a method to resolve collisions in vesicle flows. Their method requires solving the so-called nonlinear complementarity problem iteratively. The method is parameter free and allows large time steps while it guarantees no collision of vesicles.

Although solving the nonlinear problem iteratively introduces extra computational cost, they claim that it is compensated by the ability of the method to enable large time steps. In particulate flows penalty methods are often used due to their ease of implementation. These methods introduce an artificial repulsion force penalizing the close proximity of particles. The repulsion forces used in many studies [41, 42, 51, 130] are in either polynomial or exponential form. They have two parameters: One is the repulsion length scale and the other is the strength of the force. These parameters are set a priori and they are not adjusted during the simulation. As the particles come closer, increasing repulsion force leads to stiffness and results in very small time steps. In our scheme we employ a state-of-the-art penalty method from computer graphics [62, 169]. This model is in a polynomial form which performs well in dense suspension simulations because it is developed for simulations with objects coming close frequently with low velocities in the context of contact mechanics. The length scale is the only parameter of the model, which we calibrate heuristically. The strength of the repulsion is determined adaptively.

A significant question that arises in these low-resolution calculations is *an appropriate definition of the error*. Obviously one has to give up on capturing individual trajectories accurately and look at upscaled quantities such as statistics that should depend on the particular application in dense suspensions. By contrast, there are applications such as cell sorting in which the trajectories are of interest. Since we do not have a particular goal in mind and we consider this coarsening problem generically, we quantify the error in terms of individual trajectories in dilute suspensions and of upscaled quantities in dense suspensions. The dynamics and rheology of vesicle suspen-

sions have been investigated widely and various error measures have been introduced. For dilute suspensions, local error measures such error in the vesicles' inclination angles, centers and proximity to other vesicles are frequently used. In [82, 87, 99], the error is quantified using the vesicles' inclination angles and centers in dilute suspensions. In [146] distance between two vesicles in a shear flow, i.e. error in proximity. For dense suspensions, it is typical to consider collective dynamics rather than the behavior of each vesicle. For instance, effective viscosity of a suspension is an upscaling measure which is equivalent to the viscosity of a homogeneous Newtonian fluid having the same energy dissipation as the suspension [74, 146]. Additionally, in [37, 101], the so-called shear-induced diffusion, i.e., the evolution of probability distributions of vesicles' centers is investigated. This phenomenon is studied both computationally [121, 123] and experimentally [132]. We also studied mixing in vesicle suspension in [79], where we need accurate averages of velocity field. In this study, we quantify the error based on those quantities.

Organization of the Chapter In Section 2.2 we present the equations governing vesicle flows in 2D. In Section 2.3 we present the integral equation formulation of these equations and the numerical scheme. In Section 2.4 we introduce the LRCA. In Section 2.5 we test the stability of the low-resolution simulations with the LRCA in various confined and unconfined flows, and we report accuracy in terms of different error measures.

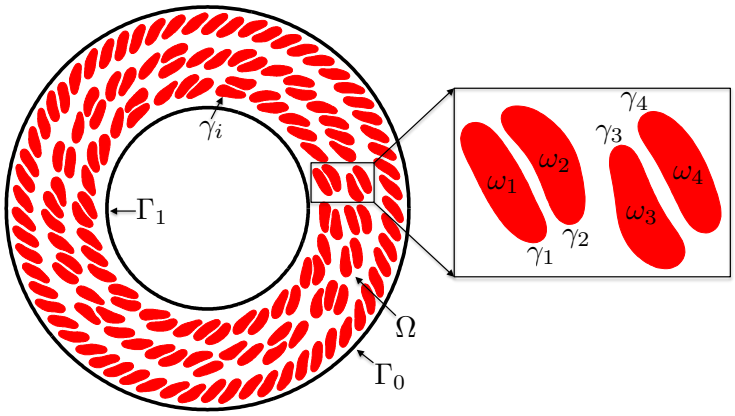


Figure 2.1: *A vesicle suspension in a Couette apparatus. Ω is the fluid domain including both the interior and the exterior of vesicles. It is bounded by fixed boundaries (walls) denoted by $\Gamma = \Gamma_0 \cup \Gamma_1$. γ_i is the boundary of the i^{th} vesicle whose interior is ω_i, $\omega = \bigcup_i \omega_i$ is the red area, and $\gamma = \bigcup_i \gamma_i$.*

2.2 Governing Equations

We use the quasi-static incompressible Stokes approximation [31, 45, 48, 180–182]. The dynamics of the flow is fully characterized by the position of the interface $\mathbf{x}(s,t) \in \gamma_i$, where s is arc-length, t is time, and γ_i is the membrane of the i^{th} vesicle. Given M vesicles, we define $\gamma = \bigcup_{i=1}^{M} \gamma_i$. The interior of the i^{th} vesicle is denoted by ω_i, and we define $\omega = \omega_1 \cup \cdots \cup \omega_M$. Let Ω be the m-ply connected domain containing the vesicles, and $\Gamma = \Gamma_0 \cup \Gamma_1 \cup \cdots \cup \Gamma_m$ be its boundary. The interior connected components of Γ are Γ_i, $i = 1, \ldots, m$, and Γ_0 is the connected component containing Γ_i. See Fig. 2.1 for the schematic of the domain. The incompressible Stokes equations describe the fluid velocity at an instance:

$$-\mu \Delta \mathbf{u}(\mathbf{x}) + \nabla p(\mathbf{x}) = 0, \quad \text{and} \quad \operatorname{div}(\mathbf{u}(\mathbf{x})) = 0, \quad \mathbf{x} \in \Omega \setminus \gamma. \tag{2.1}$$

Here, μ is dynamic fluid viscosity, $\mathbf{u}$ is the velocity and p is the pressure. We impose the velocity on the walls as a Dirichlet boundary condition

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{U}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma. \tag{2.2}$$

The no-slip boundary condition requires velocity continuity across vesicle interfaces, i.e.,

$$\mathbf{u}(\mathbf{x}, t) = \frac{d\mathbf{x}}{dt}(t), \quad \mathbf{x} \in \gamma. \tag{2.3}$$

Vesicles are inextensible, i.e., they conserve their arc-length in 2D. The inextensibility is mathematically expressed as

$$\mathbf{x}_s \cdot \mathbf{u}_s = 0, \quad \mathbf{x} \in \gamma, \tag{2.4}$$

where the subscript "s" stands for differentiation with respect to the arc-length on the boundaries of cells. Elasticity of the membrane comes into the formulation as a momentum balance on the membrane. Since vesicles resist bending and tension, their membranes apply an elastic force in response to bending and tension. The momentum balance enforces the jump in the surface traction across a membrane to be equal to the net elastic force applied by the membrane,

$$[\![\mathbf{T}\mathbf{n}(\mathbf{x})]\!] = -\kappa_b \mathbf{x}_{ssss} + (\sigma(\mathbf{x}, t)\mathbf{x}_s)_s, \quad \mathbf{x} \in \gamma, \tag{2.5}$$

where $\mathbf{T} = -p\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ is the Cauchy stress tensor, $\mathbf{n}$ is the outward normal vector on γ, $[\![\cdot]\!]$ is the jump across the membrane. The right-hand side is the net force applied by the membrane onto the fluid. The first term on the right-hand side is the force due to bending stiffness κ_b and the second term is the force due to tension σ,

which acts as a Lagrange multiplier enforcing the inextensibility [164]. Finally, the position of the boundaries of M cells evolves as

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}_\infty(\mathbf{x}_i) + \sum_{j=1}^{M} \mathbf{u}(\mathbf{x}_j), \quad i = 1, \ldots, M, \tag{2.6}$$

where $\mathbf{u}_\infty(\mathbf{x}_i)$ is the velocity induced by fixed boundaries (or the imposed velocity field in free-space flows) and $\mathbf{u}(\mathbf{x}_j)$ is the velocity due to the j^{th} vesicle acting on the i^{th} vesicle.

2.3 Integral Equation Formulation

We present the integral equation formulation of Eqns. 2.1- 2.5. Let $\nu_p = \mu_p/\mu$ denote the viscosity contrast between the fluid in the interior of the p^{th} vesicle with viscosity μ_p and the exterior fluid with viscosity μ. The single and double layer potentials for Stokes flow ($\mathcal{S}_{pq}$ and $\mathcal{D}_{pq}$, respectively) denote the potential induced by hydrodynamic densities of the interfacial force $\mathbf{f}$ and velocity $\mathbf{u}$ on vesicle q and evaluated on vesicle p:

$$\mathcal{S}_{pq}[\mathbf{f}](\mathbf{x}) := \frac{1}{4\pi\mu} \int_{\gamma_q} \left(-\mathbf{I} \log \rho + \frac{\mathbf{r} \otimes \mathbf{r}}{\rho^2} \right) \mathbf{f}(\mathbf{y}) ds_{\mathbf{y}}, \quad \mathbf{x} \in \gamma_p, \tag{2.7a}$$

$$\mathcal{D}_{pq}[\mathbf{u}](\mathbf{x}) := \frac{1-\nu_q}{\pi} \int_{\gamma_q} \frac{\mathbf{r} \cdot \mathbf{n}}{\rho^2} \frac{\mathbf{r} \otimes \mathbf{r}}{\rho^2} \mathbf{u}(\mathbf{y}) ds_{\mathbf{y}}, \quad \mathbf{x} \in \gamma_p, \tag{2.7b}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{y}$ and $\rho = \|\mathbf{r}\|_2$. Let $\mathcal{S}_p := \mathcal{S}_{pp}$ and $\mathcal{D}_p := \mathcal{D}_{pp}$ denote vesicle self-interactions. We, then, define

$$\begin{aligned} \mathcal{E}_{pq}[\mathbf{f}, \mathbf{u}](\mathbf{x}) &= \mathcal{S}_{pq}[\mathbf{f}](\mathbf{x}) + \mathcal{D}_{pq}[\mathbf{u}](\mathbf{x}), \quad \mathbf{x} \in \gamma_p, \\ \mathcal{E}_p[\mathbf{f}, \mathbf{u}](\mathbf{x}) &= \sum_{q=1}^{M} \mathcal{E}_{pq}[\mathbf{f}, \mathbf{u}](\mathbf{x}), \quad \mathbf{x} \in \gamma_p. \end{aligned}$$

For confined flows, we use the completed double layer potential due to a density function η defined on solid walls

$$\mathscr{B}[\eta](\mathbf{x}) = \mathscr{D}_\Gamma[\eta](\mathbf{x}) + \sum_{q=1}^{M} R\left[\xi_q(\eta), \mathbf{c}_q\right](\mathbf{x}) + \sum_{q=1}^{M} S\left[\lambda_q(\eta), \mathbf{c}_q\right](\mathbf{x}), \quad \mathbf{x} \in \gamma \cup \Gamma. \quad (2.9)$$

The Stokeslets and rotlets are

$$S\left[\lambda_q(\eta), \mathbf{c}_q\right](\mathbf{x}) = \frac{1}{4\pi\mu}\left(-\log\rho + \frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2}\right)\lambda_q(\eta) \quad \text{and} \quad R\left[\xi_q(\eta), \mathbf{c}_q\right](\mathbf{x}) = \frac{\xi_q(\eta)}{\mu}\frac{\mathbf{r}^\perp}{\rho^2},$$

where $\mathbf{c}_q$ is a point inside ω_q, $\mathbf{r} = \mathbf{x} - \mathbf{c}_q$, and $\mathbf{r}^\perp = (r_2, -r_1)$. The size of the Stokeslets and rotlets are

$$\lambda_{q,i} = \frac{1}{2\pi}\int_{\gamma_q} \eta_i(\mathbf{y}) ds_\mathbf{y}, \ i = 1,2 \quad \text{and} \quad \xi_q = \frac{1}{2\pi}\int_{\gamma_q} \mathbf{y}^\perp \cdot \eta(\mathbf{y}) ds_\mathbf{y}.$$

If $\mathbf{x} \in \Gamma_0$, we add the rank one modification $\mathscr{N}[\eta](\mathbf{x}) = \int_{\Gamma_0} (\mathbf{n}(\mathbf{x}) \otimes \mathbf{n}(\mathbf{y}))\, \eta(\mathbf{y}) ds_\mathbf{y}$ to $\mathscr{B}$ to remove a one-dimensional null space. Finally, we express the inextensibility constraint in operator form as

$$\mathscr{P}[\mathbf{u}](\mathbf{x}) = \mathbf{x}_s \cdot \mathbf{u}_s.$$

Therefore, the integral equation formulation is

$$\left(1 + \nu_p\right)\mathbf{u}(\mathbf{x}) = \mathscr{E}_p[\mathbf{f}, \mathbf{u}](\mathbf{x}) + \mathscr{B}_p[\eta](\mathbf{x}), \quad \mathbf{x} \in \gamma_p, \text{ vesicle evolution,} \quad (2.10a)$$

$$\left(1 + \nu_p\right)\mathbf{U}(\mathbf{x}) = -\frac{1}{2}\eta(\mathbf{x}) + \mathscr{E}_\Gamma[\mathbf{f}, \mathbf{u}](\mathbf{x}) + \mathscr{B}[\eta](\mathbf{x}), \quad \mathbf{x} \in \Gamma, \text{ fixed boundaries,} \quad (2.10b)$$

$$\mathscr{P}[\mathbf{u}](\mathbf{x}) = 0, \quad \mathbf{x} \in \gamma_p, \text{ vesicle inextensibility.} \quad (2.10c)$$

Since the velocity $\mathbf{u} = d\mathbf{x}/dt$ and the interfacial force $\mathbf{f}$ depend on σ and $\mathbf{x}$, (2.10) is a system of integro-differential-algebraic equations for $\mathbf{x}, \sigma$, and η.

2.3.1 Temporal Discretization

We discretize (2.10) in time with a first-order implicit-explicit (IMEX) time stepping method [11]. We linearize (2.10) and treat the stiff terms, such as the bending, implicitly, while treating nonlinear terms, such as the layer potential kernel, explicitly. In particular, an approximation for the position $\mathbf{x}$ and tension σ of vesicle p at the $(n+1)^{\text{th}}$ time step is computed by solving

$$\frac{\alpha_p}{\Delta t}\left(\mathbf{x}_p^{n+1}-\mathbf{x}_p^n\right) = \mathscr{S}_p^n \mathbf{f}_p^{n+1} + \mathscr{D}_p^n \mathbf{u}_p^{n+1} + \mathscr{B}_p[\eta^{n+1}] + \sum_{\substack{q=1 \\ q\neq p}}^{M} \mathscr{E}_{pq}^n[\mathbf{f}_q^{n+1}, \mathbf{u}_q^{n+1}],\ \mathbf{x}\in\gamma_p \quad (2.11a)$$

$$\mathbf{U}^{n+1}(\mathbf{x}) = -\frac{1}{2}\eta^{n+1}(\mathbf{x}) + \mathscr{E}_\Gamma^n[\mathbf{f}^{n+1}, \mathbf{u}^{n+1}](\mathbf{x}) + \mathscr{B}[\eta^{n+1}](\mathbf{x}),\ \mathbf{x}\in\Gamma, \quad (2.11b)$$

$$\mathscr{P}^n \mathbf{x}_p^{n+1} = \mathscr{P}^n \mathbf{x}_p^n,\ \mathbf{x}\in\gamma_p, \quad (2.11c)$$

$$\mathbf{u}_p^{n+1} = \frac{\mathbf{x}_p^{n+1}-\mathbf{x}_p^n}{\Delta t},\ \mathbf{x}\in\gamma_p, \quad (2.11d)$$

where $\alpha_p = (1+\nu_p)/2$, and operators with a superscript n are discretized at $\mathbf{x}^n$. Although (2.11) is fully coupled (hence computationally expensive to solve), it is more stable method than methods that treat vesicle-vesicle and vesicle-boundary interactions explicitly [142]. We solve (2.11) using GMRES with a block-diagonal preconditioner. This preconditioner removes the stiffness due to the self-interactions of vesicles but does nothing for the inter-vesicle and inter-wall interactions. As a result, the number of preconditioned GMRES iterations depends mostly on the magnitude of the inter-vesicle interactions which is a function of the vesicles' proximity. As we will see later, we upsample vesicles' boundaries to avoid aliasing. Thus, we construct the preconditioner on the upsampled grid. Although this increases the cost of building the preconditioner, the cost is offset by a significant reduction in the number of GMRES iterations.

2.3.2 Spatial Discretization

Let $\mathbf{x}(\theta)$, $\theta \in (0, 2\pi]$ be a parameterization of the interface γ_p, and let $\{\mathbf{x}(\theta_k) = 2k\pi/N\}_{k=1}^{N}$ be N uniformly distributed discretization points. Then, a spectral representation of the vesicle membrane is given by

$$\mathbf{x}(\theta) = \sum_{k=-N/2+1}^{N/2} \widehat{\mathbf{x}}(k) e^{ik\theta}.$$

We use the fast Fourier transform to compute $\widehat{\mathbf{x}}$, and arc-length derivatives are computed pseudospectrally. Nearly singular integrals are computed with an interpolation scheme [142]. Finally, we use a Gauss-trapezoid quadrature rule [5] with accuracy $\mathcal{O}(h^8 \log h)$ to evaluate the single layer potential (2.7a) and the spectrally accurate trapezoid rule for the double layer potential (2.7b).

2.4 Low-resolution Correction Algorithms

In this section, we present our low-resolution correction algorithms for simulations of vesicle suspensions: anti-aliasing in Section 2.4.1, adaptive time stepping in Section 2.4.2, local correction to area and arc-length in Section 2.4.3, reparameterization in Section 2.4.4, alignment of shapes in Section 2.4.5, and repulsion force in Section 2.4.6. In Algorithm 1, we list the order that these algorithms are called in conjunction with the advancing the vesicles forward one time step.

At every time step, we solve (2.11) with our anti-aliasing algorithm to update the vesicles' position $\mathbf{x}$, tension σ, (and density η if the flow is confined). After solving the evolution equation, given a tolerance ρ_{AL} `newTimeStepSize` determines if the solution $\mathbf{x}_{n+1}$ is accepted or rejected, and chooses a new time step size, Δt_{new}. If the

Algorithm 1 `Ves2D`:Main stages in one time step of vesicle flows

$[\mathbf{x}_{n+1}, \sigma_{n+1}, \eta_{n+1}] = \texttt{timeStep}(\mathbf{x}_n, \sigma_n, \eta_n, \Delta t_n)$ `Solve the system of equations (2.11)`
$[\texttt{accept}, \Delta t_{\text{new}}] = \texttt{newTimeStepSize}(\mathbf{x}_{n+1}, \mathbf{x}_n, \Delta t_n, T_{\text{CPU}})$ `Choose the new time step size`
if `accept` **then** `If solution is accepted`
  $\tilde{\mathbf{x}}_{n+1} = \texttt{correctShape}(\mathbf{x}_{n+1}, A_0, L_0)$ `Correct errors in area and length of vesicles`
  $\tilde{\mathbf{x}}_{n+1} = \texttt{reparameterize}(\tilde{\mathbf{x}}_{n+1})$ `Reparameterize vesicles' membranes`
  $\mathbf{x}_{n+1} = \texttt{alignShape}(\tilde{\mathbf{x}}_{n+1}, \mathbf{x}_{n+1})$ `Align reparameterized shapes with the original ones`
  $t = t + \Delta t_n$
  $\Delta t_{n+1} = \Delta t_{\text{new}}$ `Set the time step size for the next time step`
else `If solution is not accepted`
  $[\mathbf{x}_{n+1}, \sigma_{n+1}, \eta_{n+1}] \leftarrow [\mathbf{x}_n, \sigma_n, \eta_n]$ `Reject solution and try again with smaller time step`
  $\Delta t_n = \Delta t_{\text{new}}$ `Set the new time step size for the subsequent attempt`
end if

solution $\mathbf{x}_{n+1}$ is accepted, we correct the errors in area and length of every vesicle. We, then, reparameterize the vesicles' boundaries to redistribute points such that high frequency components of the surface parameterization are minimized. The reparameterization and the area-length correction cause vesicles to translate and rotate, so we align their centers and inclination angles with those of the original ones. Finally, if we detect that too much error has been committed, then the solution $\mathbf{x}_{n+1}$ is rejected and a time step is taken with a smaller time step size.

We list and comment on the parameters required by the algorithms under the pertinent sections. As a result of numerical experiments we heuristically decide on the values of these parameters. There are two main parameters setting resolution of a simulation: Spatial resolution is determined by numbers of points per vesicle N and per wall N_{wall} and the tolerance for the error in area and length at each time step, ρ_{AL}, sets the temporal resolution. At low resolutions, we have observed that spectral deferred correction (SDC) [144] does not achieve high-order accuracy in time stepping unless a very small time step is taken meaning that a small tolerance ρ_{AL} is requested.

Since we are not interested in taking small time step sizes, we do not use SDC sweeps for low-resolution simulations, but they are used for our ground truth high-resolution simulations.

We propose a black-box solver using Algorithm 1 which requires a single parameter: allocated CPU time T_{CPU} in which a simulation is desired to be completed. Our experiments in Section 2.5 show that the temporal resolution ρ_{AL} required for accurate and efficient simulations does not vary much at low spatial resolutions. The low-resolution simulations can be successfully completed using $\rho_{\text{AL}} = 10^{-2}$ or 10^{-3}. Since the errors in area and length are large at the coarse spatial resolutions, the smaller temporal resolutions result in excessive computing times at the coarse spatial resolutions, i.e. $N \leq 24$. This renders the low-resolution simulations impractical. Therefore, we do not require the temporal resolution to be defined in our solver and instead use the tolerances we consider workable at low resolutions.

Our solver starts with a coarse spatial discretization $N = 8$ points per vesicle and a high tolerance $\rho_{\text{AL}} = 10^{-2}$. Then it indicates possible refinement of the resolutions to provide an accurate physics or to avoid the failure of the simulation due to the computation time going beyond the allocated time T_{CPU}. We summarize the scheme as follows:

1. First, the solver runs the simulation with $N_0 = 8$ and $\rho_{\text{AL}} = 10^{-2}$ and monitors on-the-fly if the simulation can be completed within T_{CPU}.

2. If the estimated CPU time goes beyond the allocated time T_{CPU}, the solver terminates the simulation and increases the temporal resolution, first. The next

simulation is run with $\rho_{\mathrm{AL}} = 10^{-3}$.

3. If the estimated CPU time again exceeds the allocated time T_{CPU}, it increases the spatial resolution to $1.5N_0$ and uses $\rho_{\mathrm{AL}} = 10^{-2}$ for the next simulation.

4. The last two steps are repeated until the simulation is completed within T_{CPU}. If this is not possible, it seems that T_{CPU} is not achievable at the low resolutions.

5. Once the solver finds a resolution N and ρ_{AL}, it then checks the accuracy of the simulation. To do so, it runs two more simulations: one with $1.5N$ and $0.1\rho_{\mathrm{AL}}$, and the other with $2N$ and $0.1\rho_{\mathrm{AL}}$.

6. The self-error is computed with respect to these higher resolution simulations in terms of the quantity of interest. If the self-convergence is achieved, the simulation is terminated. If not, this means that the errors are large, so we need to increase the spatial resolution. That is, the procedure above is repeated from step 1 with larger N_0 and $\rho_{\mathrm{AL}} = 10^{-2}$.

This scheme can guarantee the accuracy of the physics in terms of the quantity of interest using as coarse discretization as possible. But it may not find the simulation which takes the shortest CPU time. However, it is expected to be faster than to simulate using some high spatial and temporal resolutions at which it is still unknown if the simulation is stable or not beforehand. Additionally, another simulation of a similar CPU time is still needed to estimate the accuracy of that solution. We test the proposed solver with an example of a microfluidic device for cell sorting in Section 2.5.7.

2.4.1 Anti-aliasing

When representing periodic functions at N grid points, only N frequencies can be represented. Therefore, if a certain operation such as the multiplication of two periodic functions is performed, new high-frequency components are formed and can not be represented with N points. These newly introduced high-frequency components are identical to one of the low-frequency components, and the result is that the high-frequency components are aliased as one of the N frequencies.

In vesicle suspensions, two operations that result in aliasing errors, especially at low resolutions, are computing the traction jump $-\kappa_b \mathbf{x}_{ssss} + (\sigma \mathbf{x}_s)_s$, and computing the single and double layer potentials (2.7). The bending term $\mathbf{x}_{ssss}$ is especially susceptible to aliasing errors since it requires multiplication by the Jacobian four times. We control the aliasing error by upsampling (uniformly). But how much should we upsample? We adjust the upsampling rate using the decay of the spectrum of $\mathbf{x}_{ssss}$. First, we upsample the N point vesicle to $16N$ points and compute the fourth derivative of this upsampled shape. Then, we systematically compare the high-frequency and low-frequency energy using a growing number of points of this upsampled shape. We start by considering the first $1.5N$ Fourier modes. If the low-frequency energy exceeds the high-frequency energy, then we use 1.5 as the upsampling rate. Otherwise, we continue by comparing the low-frequency and high-frequency energy of the first $2N$ Fourier modes. This algorithm is continued until the low-frequency energy exceeds the high-frequency energy, or the maximum upsampled rate of 16 is reached. The algorithm is outlined in Algorithm 2.

While the upsampling rate may be as large as 16, the vesicle shape is only tracked at the low resolutions with N points. Therefore, the additional cost of comput-

Algorithm 2 Choose upsampling rate for computing traction jump

Require: $\mathbf{x}$
  // Input current configuration $\mathbf{x}$
  $\mathbf{x} \leftarrow$ upsample $\mathbf{x}$ — `Upsample by a pre-specified rate of 16`
  $\mathscr{B}\mathbf{x} = \texttt{fourthDeriv}(\mathbf{x})$ — `Compute the fourth arc-length derivative of the upsampled shape`
  $\widehat{\mathscr{B}\mathbf{x}} = \texttt{fft}(\mathscr{B}\mathbf{x})$ — `Compute the FFT of the arc-length derivative`
  $\alpha = 1.5$ — `Upsample by at least 1.5`
  $\texttt{low\_energy} = \|\widehat{\mathscr{B}\mathbf{x}}(1 : \alpha N/2)\|$ — `Energy in low frequencies`
  $\texttt{high\_energy} = \|\widehat{\mathscr{B}\mathbf{x}}(\alpha N/2 + 1 : \alpha N)\|$ — `Energy in high frequencies`
  while ($\texttt{high\_energy} > \texttt{low\_energy}$ & $\alpha \leq 16$) **do**
    $\alpha = \alpha + 0.5$ — `Increase the upsampling rate`
    $\texttt{low\_energy} = \|\widehat{\mathscr{B}\mathbf{x}}(1 : \alpha N/2)\|$ — `Energy in low frequencies`
    $\texttt{high\_energy} = \|\widehat{\mathscr{B}\mathbf{x}}(\alpha N/2 + 1 : \alpha N)\|$ — `Energy in high frequencies`
  end while
  return α

ing the traction jump with our anti-aliasing algorithm is proportional to the upsampling rate. In addition, our numerical examples never required an upsampling rate larger than 10, and, at most time steps, they do not exceed 3.

In Fig. 2.2 we use Algorithm 2 to compute the aliasing error in the traction jump of a single elliptical vesicle. To compute the error, we first compute a reference traction jump with 1024 points. Then, we compute the traction jump, but with $N = 12, 16, 24$, and, 32 points both with (red) and without (blue) anti-aliasing. As expected, smaller values of N require a larger upsampling rate. In addition, the error of the Fourier modes of the traction jump when our upsampling algorithm is applied is bounded in the interval $[10^{-6}, 10^{-4}]$ for all four values of N; in contrast, when no upsampling is applied, the error decays in the low frequencies as N is increased, but remains large in the high frequencies. Finally, even when a high resolution such as $N = 32$ is used, we see that it is important to upsample by at least 1.5 to control the aliasing error.

For the layer potentials, applying Algorithm 2 is too expensive. Even if we used

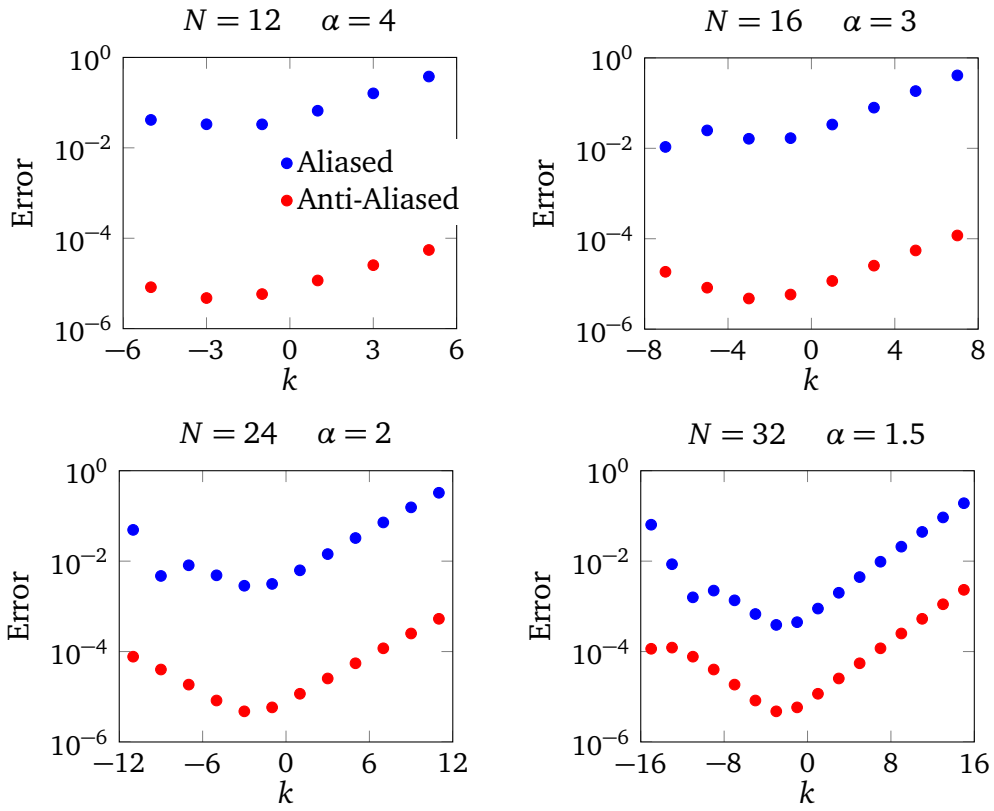


Figure 2.2: *Aliasing error of the traction jump both with and without upsampling at different resolutions. With our anti-aliasing algorithm, the aliasing error is controlled and mesh-independent. Because of symmetry in the geometry, all the even indexed Fourier modes vanish.*

a low resolution such as $N = 12$, this would require a dense matrix-vector multiplication with 192 points. Therefore, we simply fix an upsampling rate that is used at all resolutions. We have experimented with upsampling by a factor of 2 and upsampling by a factor of $\lceil \sqrt{N} \rceil$. We use the latter value since we have found that the additional cost is offset by the number of rejected time steps in some of our numerical examples. In Fig. 2.3, we plot aliasing errors with and without upsampling, again for an ellipse,

and the density function is the vesicle shape. By upsampling to $N^{3/2}$, the error is controlled at all frequencies for all the resolutions. Moreover, the upsampling rate used is less than 6 for the four small values of N that we will be considering.

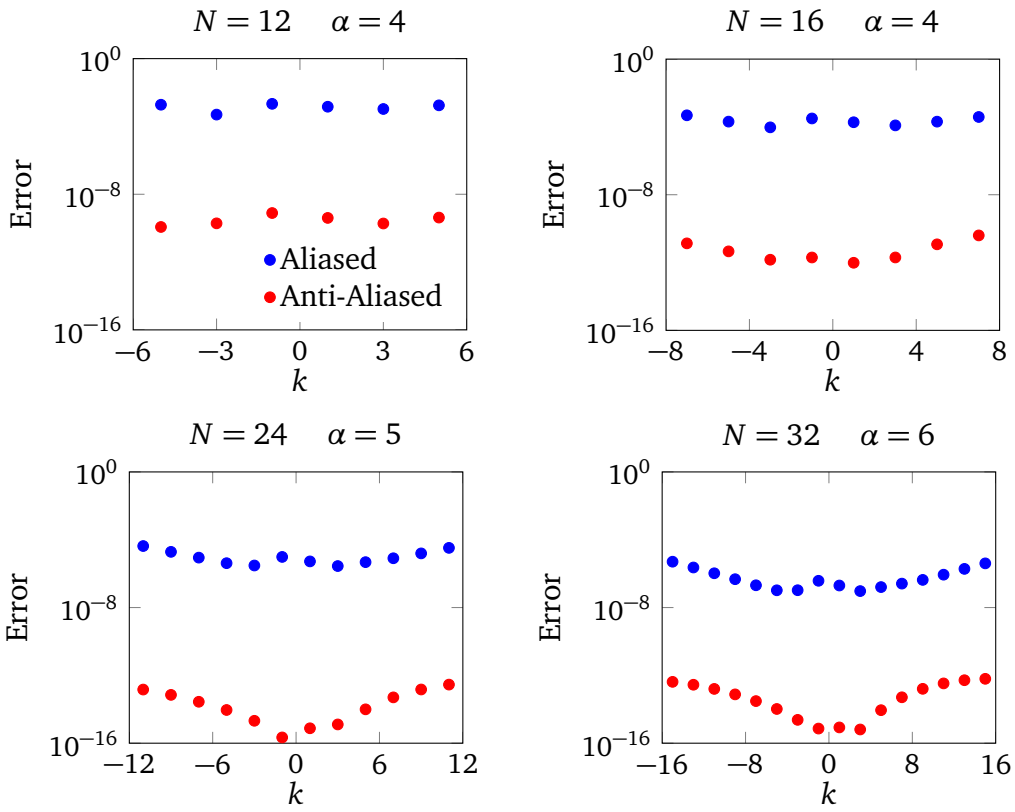


Figure 2.3: *Aliasing error of the single layer potential both with and without upsampling at different resolutions. With our anti-aliasing algorithm, the aliasing error is controlled. Because of symmetry in the geometry, all the even indexed Fourier modes vanish.*

2.4.2 Adaptive Time Stepping

The adaptive time stepping scheme introduced in [144] poses two issues. One issue is that it uses asymptotic estimates of the error, so it assumes that the temporal error dominates the spatial error, and that Δt is sufficiently small. The time stepping error does not always dominate in low-resolution simulations, and even if it does, it is possible that a very small Δt is necessary to be in the asymptotic regime. Therefore, before adjusting the time step size, we check if we are in the asymptotic regime. If we are, we use the scheme proposed in [144], and if not, then we simply increase or decrease the time step size by a constant factor. Moreover, we do not expect to achieve second- or higher-order accuracy in time, and this must be accounted for when adjusting the time step size. The second issue is that the scheme assumes accumulation of errors in area and length. However, to maintain stability, we will be correcting these errors at every time step. This is easily resolved by specifying a error tolerance for each time step rather than for the time horizon as done in [144].

We, now, describe our new scheme that uses errors in area and length to accept or reject a solution and selects a new time step size. We are going to introduce several parameters to be used in our scheme. Their values are determined as a result of several numerical experiments in a way that the number of rejected time steps is reduced. Let ρ_{AL} be the user-defined tolerance for errors in each vesicle's area and length. It sets the temporal resolution because we adjust the time step based on the area-length errors. The area A and length L of a vesicle at time t whose boundary is $\mathbf{x}(\theta, t) =$

$(x(\theta,t),y(\theta,t))$ is

$$A=\frac{1}{2}\int_0^{2\pi}(xy_\theta-yx_\theta)\,d\theta,\quad L=\int_0^{2\pi}\sqrt{x_\theta^2+y_\theta^2}d\theta.$$

Shortly we will require dA/dt and dL/dt to adjust the time step. The time derivatives are given by

$$\frac{dA}{dt}=\frac{1}{2}\int_0^{2\pi}(uy_\theta+xv_\theta-vx_\theta-yu_\theta)\,d\theta,\quad \frac{dL}{dt}=\int_0^{2\pi}\frac{x_\theta u_\theta+y_\theta v_\theta}{\sqrt{x_\theta^2+y_\theta^2}}d\theta,$$

where $u=\frac{dx}{dt}$ and $v=\frac{dy}{dt}$. We approximate the velocities with

$$u(t)=\frac{x(t+\Delta t)-x(t)}{\Delta t},\quad v(t)=\frac{y(t+\Delta t)-y(t)}{\Delta t}.$$

Suppose we compute the solution at time $t+\Delta t$ with the first-order time stepping scheme and the solution $\mathbf{x}(t+\Delta t)$ has area $A(t+\Delta t)$ and length $L(t+\Delta t)$. The errors in area and length are

$$\epsilon_A=\frac{|A(t+\Delta t)-A(t)|}{A(t)},\quad \epsilon_L=\frac{|L(t+\Delta t)-L(t)|}{L(t)}. \tag{2.12}$$

Assuming $\epsilon_A>\epsilon_L$ (the same argument holds if the situation is reversed), we either accept or reject the solution and choose a new time step size for a single vesicle (we take the maximum errors over all vesicles if we have multiple vesicles) as follows:

1. We, first, check for any collisions between different vesicles and between vesicles and solid walls using the technique presented in [142]. If there is a collision, we reject the solution and decrease the time step size to its half.

2. We define an interval $[\rho_{\text{min}},\rho_{\text{AL}}]$ where $\rho_{\text{min}}=0.5\rho_{\text{AL}}$. We accept the solution if $\rho_{\text{min}}\leq\epsilon_A\leq\rho_{\text{AL}}$, and the time step size is not changed. This helps reduce the

number of rejected time steps since it does not increase the time step size when the error is close to the tolerance ρ_{AL}.

3. If $\epsilon_A < \rho_{\text{min}}$, we check if the time step size is in the asymptotic regime. This is done by examining the Taylor series of the area

$$A(t+\Delta t) = A(t) + \frac{dA}{dt}(t)\,\Delta t + \mathcal{O}(\Delta t^2). \tag{2.13}$$

We check if the right-hand side in (2.13) is dominated by the first two terms by defining $q_A(t) = \left|\frac{dA}{dt}(t)/A(t)\right|$ so that

$$\epsilon_A \leq q_A(t)\Delta t + |\mathcal{O}(\Delta t^2)|.$$

Then, we define a tolerance ρ_{up} to determine if the asymptotic assumption can be used to adjust the time step size. That is, we say that Δt is in the asymptotic regime if

$$\frac{|\epsilon_A - q_A\Delta t|}{\epsilon_A} \leq \rho_{\text{up}}, \tag{2.14}$$

and the new time step size is

$$\Delta t_{\text{new}} = \frac{\rho_{\text{AL}}}{q_A(t)}. \tag{2.15}$$

Having ρ_{up} large leads us to use the asymptotic assumption to determine the new time step size when the assumption is not valid. Thus the number of rejected time steps increases. If condition (2.14) is not satisfied, then we increase the time step size by a constant factor β_{up}. Finally, we do not allow the time step size to exceed the maximal value Δt_{max}, which can be determined based on the length L and velocity U scales of a flow, i.e. $\Delta t_{\text{max}} \propto L/U$.

4. If $\epsilon_A > \rho_{\text{AL}}$, we reject the solution and decrease the time step size. Again, we first check if the time step size is in the asymptotic regime. For that purpose, we define another tolerance ρ_{down} which is a counterpart of ρ_{up}. If

$$\frac{|\epsilon_A - q_A \Delta t|}{\epsilon_A} \leq \rho_{\text{down}},$$

then the new time step size is chosen as in (2.15). Otherwise, we decrease the time step size by a constant factor β_{down}.

5. Once the time step size is chosen, we compute the average of the last 10 time step sizes $\overline{\Delta t}$. Then assuming that we will keep taking time steps of size $\overline{\Delta t}$ we compute the number of remaining time steps to reach the time horizon $\widetilde{m} = (T_{\text{h}} - T_{\text{current}})/\overline{\Delta t}$. We also compute the average of the CPU times it took in the last 10 time steps, $\overline{t}_{\text{CPU}}$. Then assuming that each remaining time step will take $\overline{t}_{\text{CPU}}$ on average we estimate the remaining CPU time and the total CPU time the simulation will take, $\widetilde{T}_{\text{CPU}}$. If the total estimated CPU time $\widetilde{T}_{\text{CPU}}$ exceeds the allocated time T_{CPU}, we terminate the simulation.

At low resolutions, collisions are likely as the hydrodynamic forces may not have been resolved sufficiently. In addition to the collision detection [142] in this scheme, we introduce a repulsion force in Section 2.4.6 to handle the collisions. However, an imminent collision might require small time step sizes which result in a computing time exceeding the allocated time T_{CPU}. This usually occurs when the vesicles get too close due to large time steps taken before the repulsion force is activated and once they are too close, the repulsion force introduces stiffness which requires very small

time step sizes. In those cases we terminate the simulation and take a finer temporal resolution or maybe a finer spatial resolution so that the simulation can be completed within the allocated time.

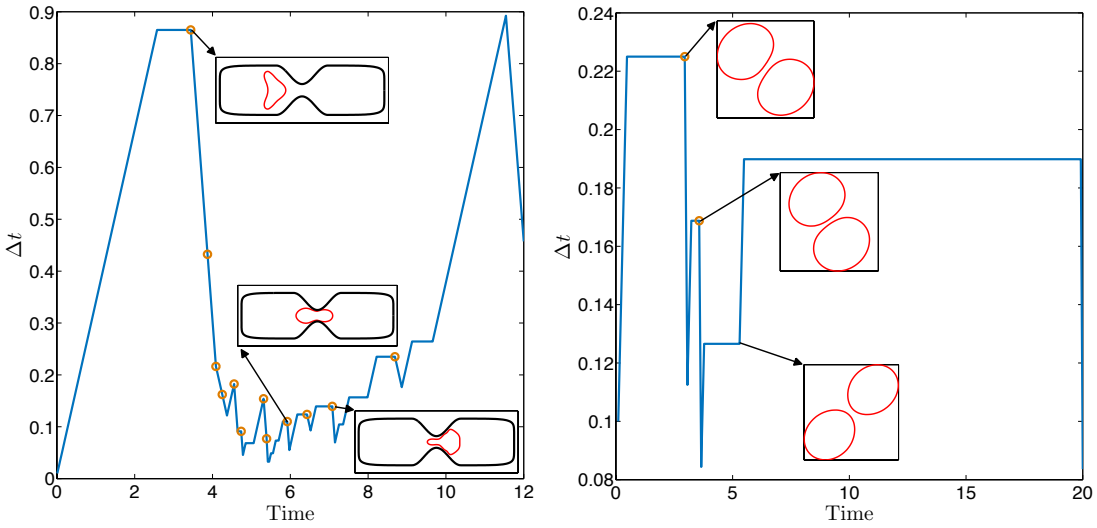
Table 2.1: *List of parameters of the adaptive time stepping scheme.*

Symbol	Definition	Value
ρ_{AL}	Tolerance for errors in area-length	[1E-4, 1E-1]
ρ_{min}	Tolerance to increase time step size	$\rho_{\text{AL}}/2$
β_{up}	Factor of increment in time step size	1.2
β_{down}	Factor of decrement in time step size	0.5
ρ_{up}	Tolerance to use asymptotic assumption	10^{-3}
ρ_{down}	Tolerance to use asymptotic assumption	10^{-2}
Δt_{max}	Maximum time step size	$\propto L/U$

We list the parameters of the adaptive time stepping and their values in Table 2.1. Here, L and U are length and velocity scales of a flow. We want to be aggressive in decreasing the time step size but cautious in increasing it. Therefore, we choose $\rho_{\text{up}} < \rho_{\text{down}}$. The other parameters are chosen by running a few experiments and choosing values that minimize the total number of rejected time steps. The parameter values in Table 2.1 work very well for a variety of problems we have tested. We demonstrate the adaptive time stepping scheme in two examples of confined and unconfined flows in Fig. 2.4.

2.4.3 Local Corrections to Area and Arc-length

The incompressibility and inextensibility conditions guarantee that the area and arc-length of each vesicle are preserved. However, long time horizon simulations suffer from the accumulation of errors in area and length which often leads to insta-



(a) Stenosis flow (b) Shear flow

Figure 2.4: *We demonstrate how the time step size varies in stenosis flow and shear flow. Open circles indicate the times when a time step size is rejected. In both simulations, vesicles are discretized with* $N = 16$ *points, and the tolerance is* $\rho_{\text{AL}} = 10^{-2}$. *In the stenosis flow, the outer wall is discretized with* $N_{\text{wall}} = 256$ *points. There are 12 rejected and 64 accepted time steps in the stenosis flow, and 2 rejected and 110 accepted time steps in the shear flow.*

bilities or nonphysical simulations. Therefore, area-length correction is essential. We introduce a postprocessing technique that maintains the errors in area and length below a prescribed tolerance without modifying the governing equations. This is done with a constrained optimization problem where the constraints require errors in area and arc-length to be below the tolerance.

Suppose that a vesicle initially has area A_0 and length L_0, and that $\mathbf{x}(t)$ is the solution at time t. We make a local correction to the vesicle's shape by applying

sequential quadratic programming (SQP) to

$$\underset{\substack{A(t)=A_0\\ L(t)=L_0}}{\operatorname{argmin}} \|\tilde{\mathbf{x}}(t)-\mathbf{x}(t)\|^2, \tag{2.16}$$

to obtain a new shape $\tilde{\mathbf{x}}$. Equation (2.16) is solved iteratively with a MATLAB built-in function, `fmincon`, which is used for minimum constrained algebraic equations (see Algorithm 3 and Algorithm 4). The function requires tolerances for the objective function ρ_{fun} and for the constraints ρ_{con}. In our low-resolution simulations, both tolerances are 10^{-3}. After correcting the area and length, it is possible that vesicles are closer than a minimum distance set by our repulsion force (see Section 2.4.6). Since we will be treating repulsion explicitly, the result would be a stiffer system and a smaller time step size would be required. To avoid this issue, we apply the correction to vesicles only if the correction does not make the distance between two vesicles less than the repulsion length scale.

We demonstrate the effectiveness of the local correction in Fig. 2.5. We consider a single vesicle of reduced area 0.65 in shear flow with no viscosity contrast. The vesicle tilts to a certain inclination angle and then undergoes a tank-treading motion. We discretize the vesicle with $N = 12$ points and reparameterize (see Section 2.4.4) its boundary at every time step. We take a time horizon of $T = 30$ so that the vesicle tank-treads approximately 1.5 times. We run the simulation with various tolerances for errors in area and length ρ_{AL}. We plot the maximum of the errors in area and arc-length without the correction (top row), snapshots of the vesicle configurations without (middle row) and with (bottom row) the local correction to the vesicle's shape. Without correction, the error grows to $\mathcal{O}(10^{-1})$ at the time horizon, and it is still growing.

However, the simulations remains stable and accurate, even with large tolerances, when the vesicle's shape is corrected.

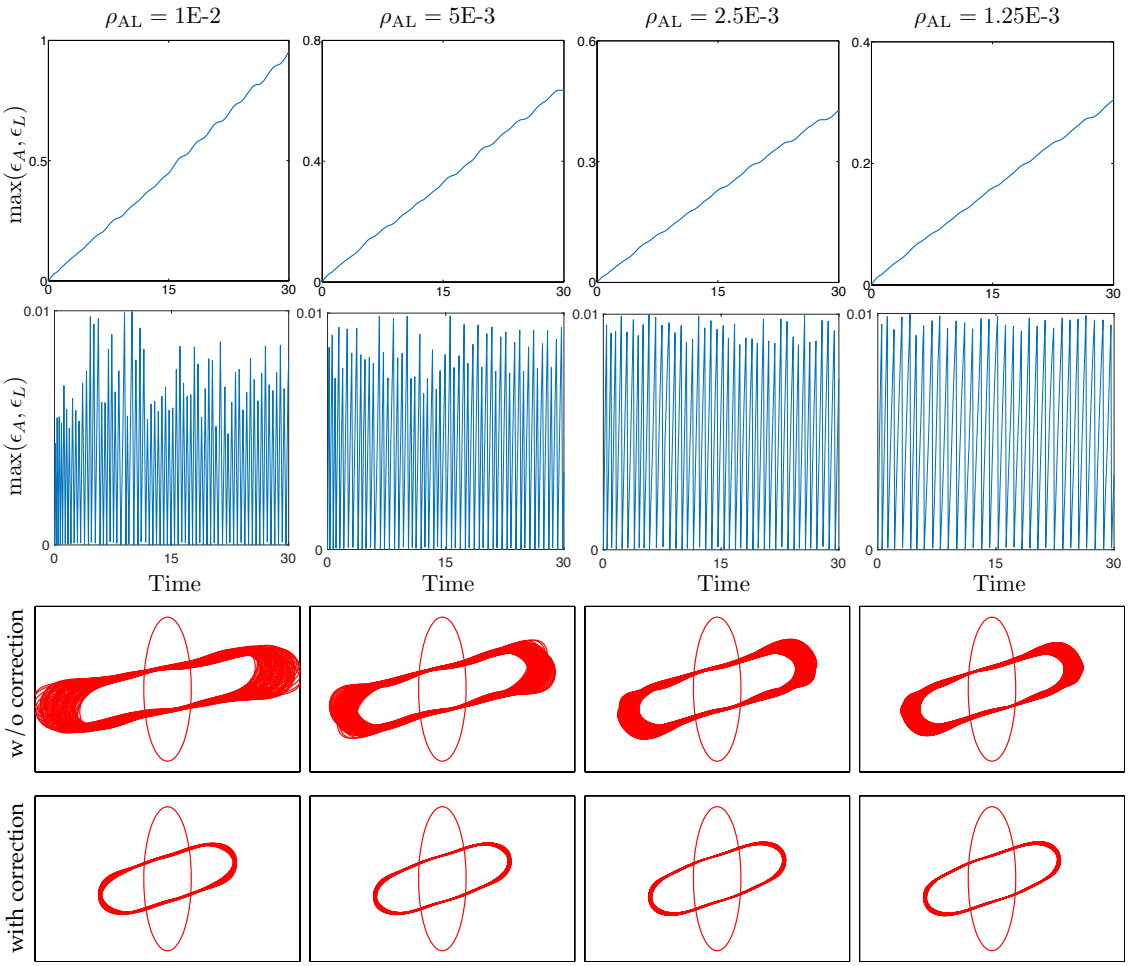


Figure 2.5: *The effect of correcting the vesicle's area and length. We discretize the vesicle with* $N = 12$ *points. Each column corresponds to a simulation with a tolerance for errors in area and length* ρ_{AL} *indicated at the top. The first and the second rows shows the maximum of the errors in area and length at each time step when the vesicle's shape is not locally corrected and corrected, respectively. The errors are corrected if they exceed the tolerance for the constraints, i.e.* $\rho_{\mathrm{con}} = 0.01$. *The third and the fourth rows shows the superimposed snapshots when the errors are not corrected and corrected at every time step, respectively.*

Algorithm 3 `correctShape`$(\mathbf{x}, A_0, L_0)$

// Input the initial area and length of each vesicle A_0, L_0
// Choose ρ_{con}, ρ_{fun} `Choose tolerances for constraints and function we want to minimize`
for $k = 1, \ldots, m$ **do** `Loop over vesicles`
  `minFun` $= @(\mathbf{z}) \|\mathbf{z} - \mathbf{x}_k\|_2$ `Minimize change in vesicle shape`
  $\tilde{\mathbf{x}}_k =$ `fmincon(minFun,...,@(z)constraints(z,`A_0, L_0`),options)` `Make a local correction`
  if `fmincon` fails **then**
    $\tilde{\mathbf{x}}_k = \mathbf{x}_k$ `If the solver fails, do not correct the` k^{th} `vesicle`
  end if
end for
`crossing = detectCollision(`$\tilde{\mathbf{x}}$`)` `Check if there is any collision of corrected shapes`
if `crossing` **then** `If there is a collision`
  $\tilde{\mathbf{x}} = \mathbf{x}$ `Do not correct any of the shapes`
else
  for $k = 1, \ldots, m$ **do** `Loop over vesicles`
    `approaching = detectNearCollision(`$\tilde{\mathbf{x}}_k$`)`
    if `approaching` **then** `if vesicles approach to each other too much`
      $\tilde{\mathbf{x}}_k = \mathbf{x}_k$ `Do not correct shape of the` k^{th} `vesicle`
    end if
  end for
end if
return $\tilde{\mathbf{x}}$

Algorithm 4 `constraints`$(\mathbf{z}, A_0, L_0)$

$A_z \leftarrow$ `getArea(z)`, $L_z \leftarrow$ `getArcLength(z)` `Compute area and arc-length of current shape z`
`cEq` $= [(A_z - A_0)/A_0 \quad (L_z - L_0)/L_0]$ `Non-linear equalities`
return `cEq` `Return constraints of the optimization problem`

2.4.4 Reparameterization

When a vesicle is discretized at low resolutions, time stepping can quickly distort the point distribution. This introduces high frequency components into the boundary parameterization which leads to aliasing errors and numerical instabilities. Therefore, it is essential to redistribute points so that high-frequency components are minimized. The reparameterization algorithm is presented in our previous work [147, 165] for three-dimensional vesicles. In Algorithm 5 we mimic this algorithm for two-

Algorithm 5 `reparameterize(x)`

Require: ρ_{y}, ρ_{g}, $\Delta\tau$,$i_{\max}$
  // We choose $\rho_{\mathrm{y}} = 10^{-3}\Delta\mathbf{x}$ and $\rho_{\mathrm{g}} = 10^{-3}$, where $\Delta\mathbf{x} = \mathbf{u}\Delta t$; and $i_{\max} = 200$
  // We use a line search to find $\Delta\tau$ at every iteration for stability
  $\mathbf{y}_0 \leftarrow$ upsample $\mathbf{x}$ `Upsample to the anti-aliasing frequency`
  $\mathbf{g}_0 = -(I - \mathbf{n}(\mathbf{y}_0) \otimes \mathbf{n}(\mathbf{y}_0))\nabla E(\mathbf{y}_0)$ `Projected gradient`
  $i = 0$
  while $i < i_{\max}$ **do**
    $\mathbf{g} = -(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y}))\,\nabla E(\mathbf{y})$ $\nabla E(\mathbf{y}) = \sum_{k=1}^{N} k^4 \widehat{\mathbf{y}}_k e^{ik\alpha}$
    $\mathbf{y}^+ = \mathbf{y} - \Delta\tau\mathbf{g}$
    $\mathbf{y} \leftarrow \mathbf{y}^+$, $i \leftarrow i + 1$
    if $\|\mathbf{g}\| < \max(\rho_{\mathrm{y}}/\Delta\tau, \rho_{\mathrm{g}}\|\mathbf{g}_0\|)$ **then**
      `break` `Terminate if the gradient or change in y is small`
    end if
  end while
  $\tilde{\mathbf{x}} \leftarrow$ downsample $\mathbf{y}$ `Downsample to the original grid`
  return $\tilde{\mathbf{x}}$

dimensional vesicles.

Let γ be the boundary of a vesicle that is parameterized as $\mathbf{x}(s)$. Let $F : \mathbb{R}^2 \to \mathbb{R}$ denote an implicit representation of the surface such that $F(\gamma) = 0$ and ∇F does not vanish. We seek a surface parameterization $\mathbf{y}(s)$ which minimizes the quality measure $E(\mathbf{y}) := \sum_{k=1}^{N} a_k |\widehat{\mathbf{y}}_k|^2$:

$$\operatorname*{argmin}_{y \in C^\infty} E(\mathbf{y}(s)), \quad \text{subject to} \quad F(\mathbf{y}(s)) = 0 \quad \text{for all} \, s,$$

where a_k are attenuation coefficients. By introducing the Lagrangian $E(\mathbf{y}) + \int_\gamma \lambda F(\mathbf{y})$, the optimality condition is obtained by taking the variation of E with respect to $\mathbf{y}$ and λ (see [165]):

$$(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y}))\,\nabla E(\mathbf{y}) = 0 \quad \text{and} \quad F(\mathbf{y}) = 0. \tag{2.17}$$

We introduce a parameter τ and use pseudo-transient continuation [91] to solve (2.17).

That is, we solve

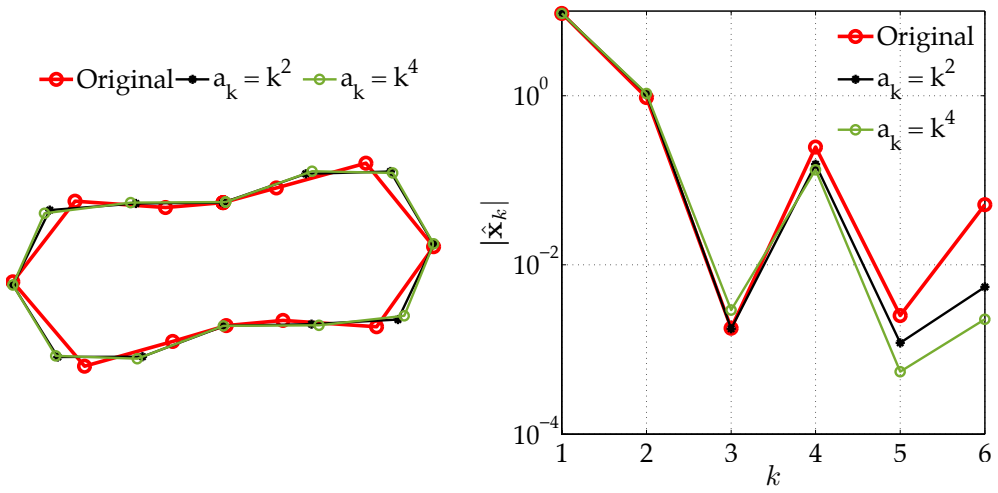
$$\dot{\mathbf{y}} + (I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y}))\nabla E(\mathbf{y}) = 0, \quad \mathbf{y}(0) = \mathbf{x}, \quad \text{and} \quad F(\mathbf{y}) = 0,$$

where $\dot{\mathbf{y}} = \partial \mathbf{y}/\partial \tau$. The discretized equation using an explicit scheme is

$$\mathbf{y}_{n+1} = \mathbf{y}_n - \Delta\tau \, (I - \mathbf{n}(\mathbf{y}_n) \otimes \mathbf{n}(\mathbf{y}_n)) \, \nabla E(\mathbf{y}_n).$$

Letting $\mathbf{g} = -(I - \mathbf{n}(\mathbf{y}) \otimes \mathbf{n}(\mathbf{y})) \, \nabla E(\mathbf{y})$, the iteration is continued until the change in $\mathbf{y}$ or the gradient $\mathbf{g}$ is sufficiently small. The parameters ρ_{y} and ρ_{g} in Algorithm 5 set this stopping criteria. Since the goal of reparameterization is to smooth the boundary γ, the attenuation coefficients a_k should be small for low frequencies and grow for high frequencies. We choose $a_k = k^4$. We have also experimented with $a_k = k^2$, but we found that the resulting shapes could still have undesirable high frequencies (see Fig. 2.6).

In Fig. 2.7, we simulate two vesicles in shear flow with and without reparameterization. The vesicles are discretized with $N = 12$ points. We use our new adaptive time stepping scheme with a tolerance of $\rho_{\mathrm{AL}} = 10^{-2}$ and we correct the area and length of the vesicles after each time step. The top row does not use reparameterization while the bottom row does. The gray vesicles are from the ground truth. The shapes with reparameterization are significantly smoother and closer to the ground truth. The number of required time steps when we reparameterize is reduced; there are 94 accepted, 4 rejected time steps with reparameterization and 108 accepted, 11 rejected time steps without reparameterization.



(a) Vesicle's shape

(b) Energies of the boundaries in Fig. 2.6(a)

Figure 2.6: *We report results for two different choices of the attenuation coefficient a_k. We reparameterize the original shape (red) discretized by $N = 12$ points in Fig. 2.6(a) with $a_k = k^2$ (black) and $a_k = k^4$ (green). The corresponding shapes are in Fig. 2.6(a). Fig. 2.6(b) shows the absolute values of the shapes' energies. While the arc-length spacing turns out to be almost uniform with $a_k = k^2$, the additional reduction in the high frequencies from using $a_k = k^4$ results in smoother vesicles and stabler simulations.*

2.4.5 Alignment of Shapes

Locally correcting and reparameterizing a vesicle shape often results in translations and rotations. In order to improve simulations' accuracy, we remove these errors after the corrections at every time step. We simply align the centers and the inclination angles (will be introduced in Section 2.5.1) of the original (obtained by solving the governing equations) and the corrected shapes.

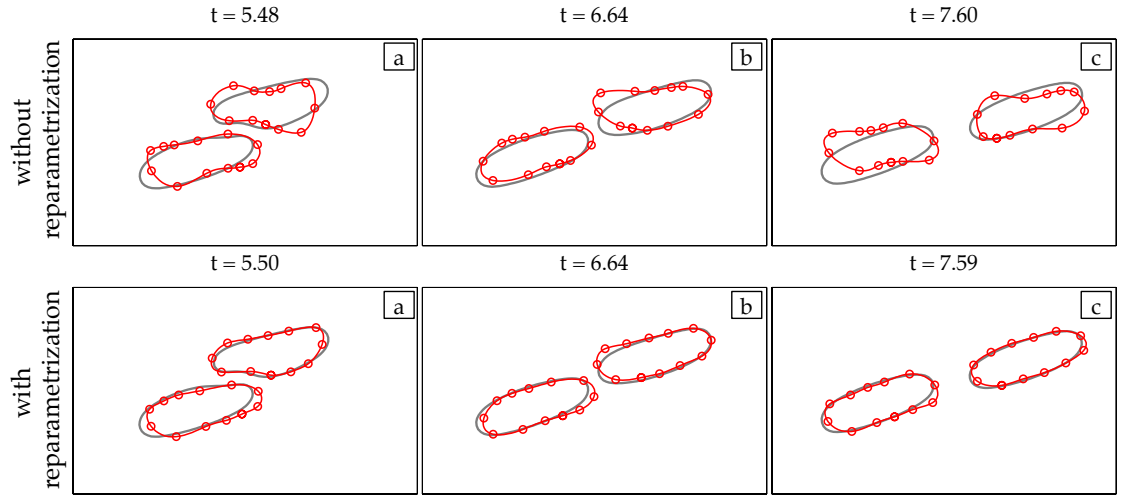
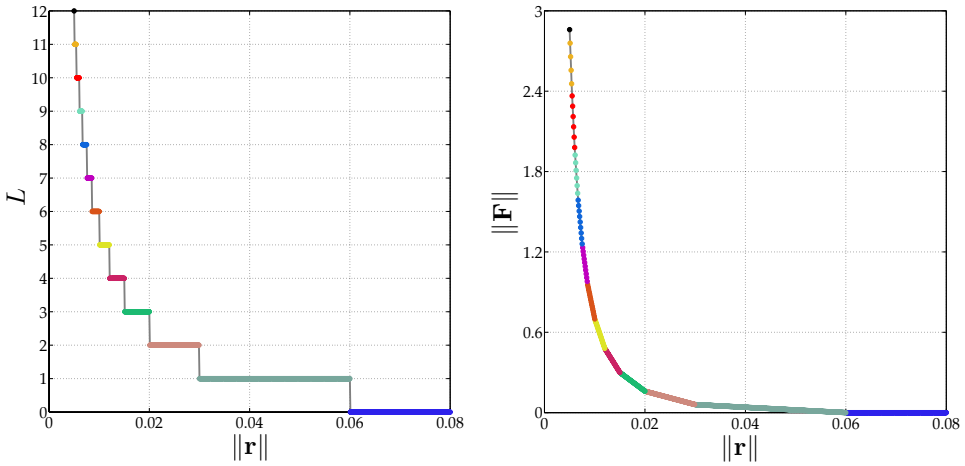


Figure 2.7: *Our reparameterization algorithm applied to two vesicles of reduced area* 0.65 *in shear flow with no viscosity contrast. The vesicles are discretized with* $N = 12$ *points and we set* $\rho_{\mathrm{AL}} = 10^{-2}$*. The top row does not use reparameterization while the bottom row does. The gray vesicles are the ground truth solution, which is computed using the high-fidelity version of the code.*

2.4.6 Repulsion

While hydrodynamic forces do not allow vesicles to collide, these forces are often not accurately resolved in simulations with low spatial resolutions, and hence vesicles may collide. We introduce a repulsion force to handle collisions. We use discrete penalty layers to penalize close proximity between discretization points on vesicles. The form of the repulsion we use has been introduced for contact mechanics [62, 169]. Letting $h_{\max}$ be the maximum arc-length spacing and $d_{\min}$ a repulsion length scale, the repulsion force applies on the points of the vesicles' membranes when they get closer than $d_{\min} = \delta_{\min} h_{\max}$. We define a gap function for discrete layer ℓ between two discretization points $\mathbf{x} \in \gamma_p$ and $\mathbf{y} \in \gamma_q$, $p \neq q$

$$g_\ell = \|\mathbf{r}\| - \frac{d_{\min}}{\ell},$$



(a) Number of activated layers vs. distance (b) Repulsion force vs. distance

Figure 2.8: *We take two approaching points and compute the total number of activated discrete layers L and the total penalty force* **F** *as discussed in* (2.19). *We choose* $d_{\min} = 0.06$ *and* $W = 1$. *We show each L on the left and magnitude of the corresponding total penalty force* $\|\mathbf{F}\|$ *on the right with the same color. The repulsion force increases as the points approach each other.*

where $\|\mathbf{r}\| = \|\mathbf{x} - \mathbf{y}\|$. The gap function measures the proximity of two points on the vesicles (γ_p and γ_q). When $g_\ell < 0$, the points are in the proximity of the layer ℓ. The repulsion force to penalize being in the proximity of the ℓ^{th} discrete layer is

$$\mathbf{F}^l = \begin{cases} -2W\ell^2 \frac{g_\ell}{\|\mathbf{r}\|}\mathbf{r}, & \text{if } g_\ell < 0, \\ 0, & \text{otherwise,} \end{cases} . \tag{2.18}$$

where W is the repulsion strength. The penalty force can be considered as placing a spring between approaching vesicles. If there is a single spring between them, the spring will compress fully and eventually fail for sufficiently large relative velocity. However, having penalty forces as a function of the active discrete layers as in (2.18) can be considered as placing an infinite number of springs between approaching vesicles. This guarantees that two vesicles do not collide, which makes the method robust.

Although this guarantee is independent of the repulsion strength W, performance of the method and error in physics depend on the choice of W. The total number of activated discrete layers, L, is the largest integer less than $\frac{d_{\min}}{\|\mathbf{r}\|}$. Hence, the total penalty force on point $\mathbf{x} \in \gamma_p$ due to point $\mathbf{y} \in \gamma_q$ is

$$\mathbf{F} = \sum_{\ell=1}^{L} \mathbf{F}^{\ell} = W \left(-\frac{L(L+1)(2L+1)}{3} + L(L+1)\frac{d_{\min}}{\|\mathbf{r}\|} \right) \mathbf{r}, \quad L = \left\lfloor \frac{d_{\min}}{\|\mathbf{r}\|} \right\rfloor \tag{2.19}$$

In Fig. 2.8 we plot the total number of discrete layers activated L and the total penalty force of two approaching points. We show each L in Fig. 2.8(a) and the corresponding total penalty force $\mathbf{F}$ in Fig. 2.8(b) with the same color. As the points approach to each other, the number of activated layers L increases and the color of the curves showing L and $\mathbf{F}$ simultaneously change. Finally, the repulsion force at a point $\mathbf{x} \in \gamma_p$ due to all other vesicles is formed by summing (2.19) over all discretization points $\mathbf{y} \notin \gamma_p$. We treat the repulsion force explicitly. That is, single layer potentials of the repulsion forces are computed and placed on the right hand side of the linear system. That can introduce stiffness when the vesicles suddenly come too close.

Remark. In order to choose the repulsion length scale $d_{\min}$, we place two vesicles of reduced area 0.65 symmetrically about the origin in an extensional flow. This simulation is done at a low resolution with $N = 12$ points. We examine the energy in the six lowest frequencies relative to the total energy of the vesicles' velocities. This ratio is used to heuristically set $d_{\min}$. In this example, when the vesicle separation is less than $0.3h_{\max}$, this ratio drops significantly and high-frequency components appear. Therefore, we set the repulsion length scale to $d_{\min} = 0.3h_{\max}$ implying that the minimum

distance between two points on the vesicles is $d_{\min} = 0.3h_{\max}$ after which the repulsion force is non-zero. However, in our experiments we observed that the vesicles got so close that an imminent collision required very small time step sizes in some cases and $d_{\min} = 0.5h_{\max}$ performs better in those cases. Therefore, we set the repulsion length scale to $d_{\min} = 0.5h_{\max}$ and never adjusted it again. We set the repulsion strength W so that the velocity induced by the repulsion force is 10% of the velocity due to all hydrodynamic forces in the example above. While vesicles can approach one another in various ways, this example represents one of the worse case scenarios since the proximity between the vesicles decreases for all time, and we have successfully used this length scale parameter for all of our experiments in Section 2.5.

2.5 Numerical Experiments

In this section, we demonstrate with various examples that the low-resolution correction algorithms are necessary to maintain stability and to improve accuracy in low-resolution simulations. We present the accuracy in terms of different error measures. The error measures are discussed in Section 2.5.1, and a summary of the numerical experiments are in Section 2.5.2.

2.5.1 Error Measures

We examine the convergence of the method to a ground truth, where the ground truth is formed at high resolutions. We also report self-error of a low-resolution simulation with respect to another low but higher resolution simulation. We denote the error with respect to a ground truth with ϵ^{g} and the self-error with ϵ^{s}. We present

two sets of error measures, one for dilute suspensions and one for dense suspensions.

For dilute suspensions, we are interested in configuration and trajectory of each vesicle. We use several error measures described as follows. Let $\widehat{\mathbf{x}}_k$ and $\mathbf{x}_k$, $k = 1,\ldots,m$, denote the position of m vesicles formed with a high-resolution simulation (either ground truth or another low-resolution simulation) and with a low-resolution simulation, respectively. If $\widehat{\mathbf{c}}_k$ and $\mathbf{c}_k$ are the centers of the vesicles, then the error over all time of the center of the k^{th}vesicle, and the maximum of this error over all vesicles are

$$\epsilon^k_{\text{center}} = \max_{t\in[0,T]} \begin{cases} \frac{\|\mathbf{c}_k - \widehat{\mathbf{c}}_k\|_2}{\varepsilon}, & \text{if } \|\widehat{\mathbf{c}}_k\|_2 = 0 \\ \frac{\|\mathbf{c}_k - \widehat{\mathbf{c}}_k\|_2}{\|\widehat{\mathbf{c}}_k\|_2}, & \text{otherwise} \end{cases}, \quad \epsilon_{\text{center}} = \max_{k=1,\ldots,m} \epsilon^k_{\text{center}}, \tag{2.20}$$

where ε is MATLAB's floating point relative accuracy eps. The error in proximity is used for examples with two vesicles. Letting $\mathbf{d} = \mathbf{c}_1 - \mathbf{c}_2$ and $\widehat{\mathbf{d}} = \widehat{\mathbf{c}}_1 - \widehat{\mathbf{c}}_2$, the error in proximity of two vesicles is

$$\epsilon_{\text{prox}} = \max_{t\in[0,T]} \frac{\|\mathbf{d} - \widehat{\mathbf{d}}\|_2}{\|\widehat{\mathbf{d}}\|_2}. \tag{2.21}$$

The inclination angle of a vesicle is the angle between the x-axis and the principal axis (eigenvector of (2.22)) corresponding to the smallest principal moment of inertia (eigenvalue of (2.22)) [146]. The moment of inertia tensor is

$$J = \int_\omega \left(|\mathbf{r}|^2 I - \mathbf{r}\otimes\mathbf{r}\right) d\mathbf{x} = \frac{1}{4}\int_\gamma \mathbf{r}\cdot\mathbf{n}\left(|\mathbf{r}|^2 I - \mathbf{r}\otimes\mathbf{r}\right) ds, \tag{2.22}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{c}$, and $\mathbf{c}$ is the center of the vesicle. Then the error over all time of the inclination angle of the k^{th} vesicle, and the maximum over all vesicles are

$$\epsilon^k_{\text{IA}} = \max_{t\in[0,T]} \begin{cases} \frac{|\text{IA}_k - \widehat{\text{IA}}_k|}{\varepsilon}, & \text{if } |\widehat{\text{IA}}_k| = 0 \\ \frac{|\text{IA}_k - \widehat{\text{IA}}_k|}{|\widehat{\text{IA}}_k|}, & \text{otherwise} \end{cases}, \quad \epsilon_{\text{IA}} = \max_{k=1,\ldots,m} \epsilon^k_{\text{IA}}. \tag{2.23}$$

For dense suspensions, the errors in the vesicles' configurations at low resolutions are large and irrelevant. However, depending on the purpose of the simulation, low-resolution simulations can provide significant information with a considerably low computational cost. We consider upscaling measures such as errors in statistics and space-time averages of physical quantities. The velocity field of the bulk fluid plays an important role in many applications. For instance, in Chapter 3 we study mixing in a Couette apparatus containing vesicles (see Fig. 2.19). We model transport with an advection-diffusion equation, so capturing the correct averages of the velocity field is crucial. We consider the error in space-time average of the velocity field and the error in time average of the L^2 norm of the velocity field. The space and space-time averages of a velocity field $\mathbf{V}(\mathbf{x},t)$ are

$$v(t)=\frac{1}{|\Omega|}\int_{\Omega}\mathbf{V}(\mathbf{x},t)d\mathbf{x},\quad \langle v\rangle=\frac{1}{T}\int_0^T v(t)dt, \tag{2.24}$$

respectively. Letting $\langle\widehat{v}\rangle$ and $\langle v\rangle$ denote the space-time averages of velocity fields given by a high-resolution simulation and its corresponding low-resolution simulations, the error is

$$\epsilon_{\langle v\rangle}=\frac{|\langle v\rangle-\langle\widehat{v}\rangle|}{|\langle\widehat{v}\rangle|}. \tag{2.25}$$

Additionally, the L^2 norm of the velocity field and the time average of this quantity are

$$V(t)=\frac{1}{|\Omega|}\Big(\int_{\Omega}\mathbf{V}^2(\mathbf{x},t)d\mathbf{x}\Big)^{\frac{1}{2}},\quad \langle V\rangle=\frac{1}{T}\int_0^T V(t)dt, \tag{2.26}$$

respectively. The error in the time average of the L^2 norm of a velocity field is

$$\epsilon_{\langle V\rangle}=\frac{|\langle V\rangle-\langle\widehat{V}\rangle|}{|\langle\widehat{V}\rangle|}. \tag{2.27}$$

Another error measure is based on a numerical homogenization for suspension rheology. The effective viscosity of a suspension is the viscosity of a homogeneous Newtonian fluid having the same energy dissipation per macroscopic volume element. For vesicle suspensions, it is given by [146]

$$\mu_{\text{eff}} = \mu + \phi \frac{1}{T} \int_0^T \overline{\sigma}^p_{12} dt \tag{2.28}$$

where

$$\overline{\sigma}^p = \frac{1}{|\Omega|} \int_\gamma \left[-\left(\kappa_b \kappa^2 \mathbf{n} \otimes \mathbf{n} + \sigma \mathbf{t} \otimes \mathbf{t} \right) + \mu(\nu - 1)\left(\mathbf{u} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{u} \right) \right] ds. \tag{2.29}$$

Here, μ is viscosity of the exterior fluid, ϕ is the area fraction of vesicles (the ratio between the area occupied by vesicles and total area), $\overline{\sigma}^p$ is the spatial average of the perturbation in stress σ^p due to the presence of vesicles, κ_b is the bending stiffness, κ is the curvature, $\mathbf{n}$, $\mathbf{t}$ are the unit normal and tangent vectors, and $\mathbf{u}$ is the velocity. Letting $\widehat{\mu}_{\text{eff}}$ and μ_{eff} be effective viscosities of a suspension obtained from a high- and a low-resolution simulation, the error in effective viscosity of a suspension is

$$\epsilon_{\mu_{\text{eff}}} = \frac{|\mu_{\text{eff}} - \widehat{\mu}_{\text{eff}}|}{|\widehat{\mu}_{\text{eff}}|}. \tag{2.30}$$

For dense suspensions in a Couette apparatus, we also report probability distribution functions of the location of each vesicle's center and the magnitude of the velocity at certain radii.

We report the *self-convergence* of the solutions within the low- resolution simulations in addition to the convergence to a ground truth solution. In the following sections, the self-errors ϵ^s are reported in a way that they are computed with respect to

the simulation whose results are in a row below on the same table. So the self-errors show how much the quantity of interest changes as we refine the resolution. In order to be able to observe the self-convergence, we need to have at least three runs with different resolutions. If the self-errors of these runs are oscillating by an order of magnitude, then the self-convergence is not achieved. If we observe self-errors slightly oscillating or decreasing as the resolution increases, then the self-convergence is attained. The numerical experiments we have conducted show that the self-error of $\mathcal{O}(10^{-2})$ is sufficient to confirm the convergence towards the ground truth. The self-convergence is useful to estimate the accuracy of a low-resolution simulation when a ground truth solution is not available.

2.5.2 Summary of Numerical Experiments

We perform numerical experiments of both dilute and dense vesicle suspensions in bounded and unbounded domains. We use our adaptive time stepping in all runs except when forming the ground truth. Then, we compare the simulations with and without the *LRCA*. We report timings and the (self-) errors defined in Section 2.5.1. A simulation is terminated if it takes orders of magnitude more computing time than the other simulations of the same example with different resolutions. The examples we consider are:

- **Two vesicles in a shear flow** (Section 2.5.3): We simulate a pair of vesicles with viscosity contrasts $\nu = 1$ and $\nu = 10$ and reduced areas 0.65 and 0.99. The initial configurations result in the vesicles nearly touching. The purpose of these experiments is to demonstrate errors in average quantities such as the proximity

between vesicles and the actual trajectories of the vesicles.

- **One vesicle in a stenosis flow** (Section 2.5.4): We simulate a single vesicle of reduced area 0.65 and without viscosity contrast $\nu = 1$ in a constricted tube (stenosis) with a parabolic flow profile at the intake and the outtake. In these experiments, the vesicle's initial height is 3.5 times larger than the constriction size. As a result of that it highly deforms and gets close to the tube's boundary as it passes the constriction. Here, we show that the LRCA are essential to avoid the vesicle-solid boundary collisions.

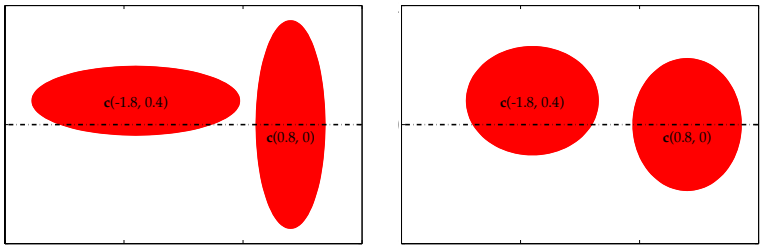
- **Four vesicles in a Taylor-Green flow**(Section 2.5.5): We simulate four vesicles of reduced area 0.65 with viscosity contrasts of $\nu = 1$ and $\nu = 10$ in a periodic Taylor-Green flow. The vesicles cover approximately 50% of the area of a periodic cell $(0, \pi)^2$ making vesicle interactions stronger and the problem more complicated than the previous examples. Here we demonstrate that although the simulations do not converge in terms of the local error measures such as ϵ_{center}, the convergence in the upscaling measures can be achieved at low resolutions.

- **Couette apparatus** (Section 2.5.6): We simulate vesicles of reduced area 0.65 without viscosity contrast in a Couette apparatus. Simulations with area fractions $\phi = 20\%$ and $\phi = 40\%$ are performed. For these examples, we report errors in the upscaled quantities and statistics. Similar to the experiment with a Taylor-Green flow, many vesicle interactions result in large local errors. However, the low-resolution simulations are 10×-100× faster while capturing the upscaled quantities and statistics accurately.

- **Microfluidic device** (Section 2.5.7): We simulate the separation of a healthy red blood cell (RBC) in a microfluidic device using deterministic lateral displacement (DLD) technique [71]. The device we consider here leads the RBC to show no net lateral displacement, which is confirmed by the actual and numerical experiments [19, 97]. The purpose of this experiment is to show the ability of our black-box solver to deliver the accurate physics using as coarse discretization as possible.

Remark. For all runs, we fix the bending stiffness to $\kappa_b = 10^{-1}$ and the GMRES tolerance to $\rho_{\text{GMRES}} = 10^{-10}$. Ground truth solutions computed with the high-fidelity version of the code are illustrated as gray vesicles. Additionally, since we use our adaptive time stepping scheme, simulations are compared at different, but comparable times.

2.5.3 Shear Flow



(a) RA = 0.65 (b) RA = 0.99

Figure 2.9: *Initial configurations of two vesicles in free-space shear flow. Vesicles in (a) are of reduced area* 0.65 *and those in (b) are of reduced area* 0.99*. For both configurations we perform simulations with viscosity contrasts* $\nu = 1$ *and* $\nu = 10$*.*

Table 2.2: *Parameters of the ground truth for shear flow.*

Parameter	Value
Points per vesicle N	96
Viscosity contrast ν	$\{1, 10\}$
Number of SDC sweeps n_{sdc}	1
Time step size Δt	5×10^{-4}
CPU time (RA $= 0.65$, $\nu = 1$)	17 hours
CPU time (RA $= 0.65$, $\nu = 10$)	61 hours
CPU time (RA $= 0.99$, $\nu = 1$)	16 hours
CPU time (RA $= 0.99$, $\nu = 10$)	36 hours

Setup. We consider two vesicles in a free-space shear flow $\mathbf{u} = (y, 0)$. The initial configuration (Fig. 2.9) results in the left vesicle traveling to the right and over top of the right vesicle. We consider reduced areas 0.65 and 0.99 and viscosity contrasts $\nu = 1$ and $\nu = 10$. We simulate each of these cases with $N = 12, 16, 24, 32$ points per vesicle and an error tolerance $\rho_{\text{AL}} = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ with and without the LRCA. The time horizon is $T = 20$ so that the vesicles pass one another. The ground truth solutions are formed with the parameters in Table 2.2.

Results. We investigate the necessity of the LRCA to maintain stability and we quantify their effect on the error in the proximity of the vesicles, ϵ_{prox}. This problem is particularly difficult because the hydrodynamic force is inaccurate at low resolutions, and this can cause vesicles to collide. We report the (self-) errors in proximity, the number of accepted and rejected time steps, and the CPU times.

In Table 2.3, we summarize the simulations of two vesicles of reduced area 0.65 with $\nu = 1$ (top) and $\nu = 10$ (bottom). For almost all the simulations, the LRCA are not necessary to maintain stability. However, the error in the proximity of the

vesicles is decreased when the LRCA are used for all runs except $N = 32$ with $\nu = 1$, and for the two highest resolutions with $\nu = 10$. In these cases where the simulations with the LRCA have greater errors in the proximity than the original simulations, the resolution is sufficient for stability without the LRCA and the effects of the LRCA do not vanish yet, i.e. the repulsion length scale is greater than the closest distance between vesicles accurately captured at those resolutions. That is why the original simulations are more accurate than the ones with the LRCA. In addition, as expected, the CPU time increases when the algorithms are used, but the payoff is additional stability and accuracy in almost all the examples. We also increase the temporal resolution while keeping the spatial resolution the same for $N = 12$ and $N = 16$. By doing so, the errors in the proximity become less than those given by the two highest spatial resolutions in shorter CPU times. However, lowering the tolerances at the coarse spatial resolutions might significantly increase the number of time steps taken and hence the CPU time because it requires small time steps to keep the errors in area and length below those low tolerances. Therefore, it is not always efficient to refine the temporal resolution only. For example, in the shear flow of two vesicles with RA $= 0.65$ and $\nu = 1$ decreasing the tolerance from $\rho_{\mathrm{AL}} = 10^{-2}$ to $\rho_{\mathrm{AL}} = 10^{-3}$ with $N = 12$ leads to a sixfold increase in the CPU time (see Table 2.3). Yet increasing the spatial resolution from $N = 12$ to $N = 16$ while decreasing the tolerance only triples the CPU time and results in a smaller error. Here both the error with respect to the ground truth and the self-error decrease as the spatio-temporal resolution is increased. So we observe that the self-convergence within the low-resolution simulations indicates the convergence towards the ground truth.

Fig. 2.10 shows snapshots of the simulation without viscosity contrast at a particular resolution both with and without LRCA, and it is clear that the LRCA are necessary to maintain physical vesicle shapes. So, the error metric we are using seems to be underestimating the error. Although the error in proximity can be considered reasonable, the original simulation has nonphysical vesicles. The alternative error measure here would be $\max(\epsilon_{\text{prox}}, \epsilon_A, \epsilon_L)$, i.e. the maximum of the errors in proximity, area and length. For the simulations with the LRCA, this error measure would be equivalent to ϵ_{prox} for low resolutions if the tolerance for the area-length correction is small (i.e. the errors in area-length are small). In Fig. 2.11 the two vesicles with $\nu = 10$ are illustrated at three different resolutions with the LRCA.

Table 2.3: *The (self-) errors in the proximity of two vesicles of RA* $= 0.65$ *with viscosity contrast* $\nu = 1$ *(top) and* $\nu = 10$ *(bottom) in the shear flow with and without the LRCA (see Figures 2.10 and 2.11). The self-errors are computed with respect to the simulation in one row below. Also reported are the number of accepted and rejected time steps and the CPU time. The dash "-" is put on the table for the simulations which break without the LRCA because the vesicle collisions cannot be handled. The ground truth simulations of* $\nu = 1$ *and* $\nu = 10$ *take 17 and 61 hours, respectively. Both simulations have* $N = 96$ *and* $\Delta t =$ *5E-4.*

$\nu = 1$

		LRCA					Original			
N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{prox}}$	$\epsilon^{\text{s}}_{\text{prox}}$	Accepts	Rejects	Time (sec)	$\epsilon^{\text{g}}_{\text{prox}}$	Accepts	Rejects	Time (sec)
12	1E-2	1.6E-1	1.1E-1	94	4	64	2.5E-1	128	10	58
16	1E-3	5.7E-2	4.5E-2	310	7	193	2.0E-1	345	14	132
24	1E-4	4.4E-2	1.7E-2	998	11	826	4.1E-2	1026	14	400
32	1E-5	3.2E-2		3156	15	1930	9.1E-3	3174	15	808
12	1E-2	1.2E-1	1.0E-1	94	4	64	2.5E-1	128	10	58
12	1E-3	8.0E-2	6.7E-2	354	10	348	8.4E-2	761	10	335
12	1E-4	1.5E-2		1165	15	1150	1.0E-2	5226	16	1880
16	1E-3	3.7E-2	3.5E-2	310	7	193	2.0E-1	345	14	132
16	1E-4	2.6E-2		955	5	921	1.9E-1	2086	21	889

$\nu = 10$

		LRCA					Original			
N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{prox}}$	$\epsilon^{\text{s}}_{\text{prox}}$	Accepts	Rejects	Time (sec)	$\epsilon^{\text{g}}_{\text{prox}}$	Accepts	Rejects	Time (sec)
12	1E-2	3.9E+0	2.3E+0	93	9	60	3.0E+0	98	11	41.4
16	1E-3	1.1E+0	1.5E+0	227	19	205	-	-	-	-
24	1E-4	3.1E-1	3.9E-1	786	32	843	1.6E-1	773	31	438
32	1E-5	1.4E-1		2567	37	2660	3.7E-2	2480	35	1120
12	1E-2	3.9E+0	3.2E+0	93	9	60	3.0E+0	98	11	41.4
12	1E-3	6.4E-1	1.4E+0	274	11	366	-	-	-	-
12	1E-4	7.4E-2		844	12	1220	-	-	-	-
16	1E-3	1.1E+0	2.1E-1	227	19	205	-	-	-	-
16	1E-4	4.7E-1		789	11	1530	-	-	-	-

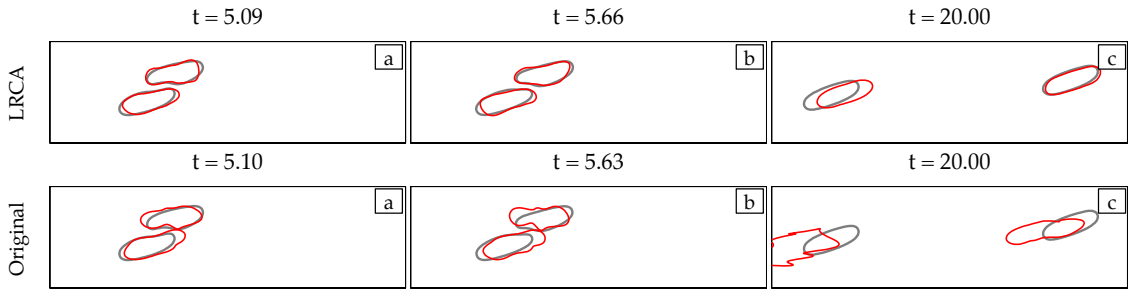


Figure 2.10: *Two vesicles with reduced area* 0.65*, viscosity contrast* $\nu = 1$*, and discretized with* $N = 12$ *points in shear flow. The error tolerance is* $\rho_{\mathrm{AL}} = 10^{-2}$*. In the top row, the LRCA are used, and in the bottom row, they are not. The error metric we are using seems to be underestimating the error. Although the error in proximity can be considered reasonable, the original simulation has nonphysical vesicles.*

Finally, we present results for the vesicles of reduced area 0.99 with the two different viscosity contrasts in Table 2.4. Here, vesicles do not come as close as those of reduced area 0.65. At all the resolutions we consider, not using the LRCA delivers more accurate results in terms of the vesicles' proximity with less CPU time. However if the algorithms are not used at the resolutions $N \leq 16$, the errors in area and length of the vesicles are $\mathcal{O}(10^{-1})$. This leads the vesicles to have nonphysical shapes at the time horizon (see Fig. 2.12 and Fig. 2.13). As the spatio-temporal resolution increases, the error with respect to the ground truth decreases. However, the self-error does not monotonously decrease for $\nu = 1$. For example, the self-error of ($N = 16$, $\rho_{\mathrm{AL}} = 10^{-3}$) is higher than that of ($N = 12$, $\rho_{\mathrm{AL}} = 10^{-2}$). This means that the quantity of interest still changes significantly as the resolution increases from ($N = 16$, $\rho_{\mathrm{AL}} = 10^{-3}$) to ($N = 24$, $\rho_{\mathrm{AL}} = 10^{-4}$). When the resolution is increased further, the self-error decreases an order of magnitude. In other words, the quantity of interest does not change much anymore. So the self-convergence is observed. This agrees well with the error with

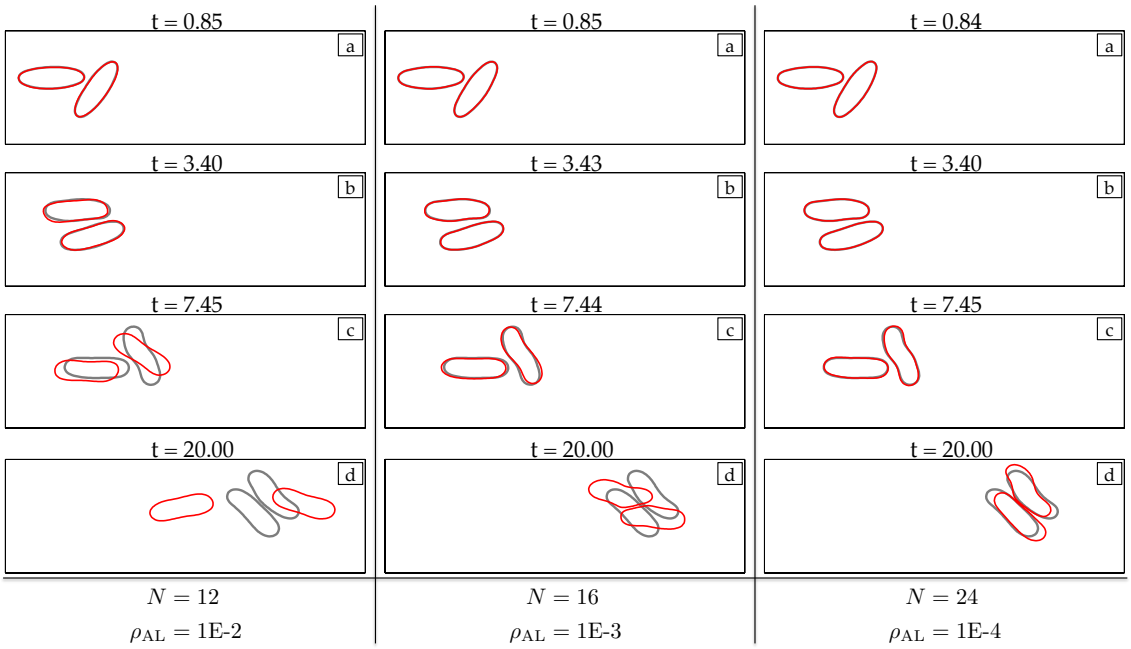


Figure 2.11: *Two vesicles with RA = 0.65 and $\nu = 10$ in shear flow using the LRCA. Vesicles are discretized with $N = 12, 16, 24$ points and the error tolerances in area and length are $\rho_{\text{AL}} = 10^{-2}, 10^{-3}, 10^{-4}$, respectively. We show that as the resolution increases, the low-resolution simulations converge among themselves and to the ground truth.*

respect to the ground truth. As the resolution is increased from ($N = 16$, $\rho_{\text{AL}} = 10^{-3}$) to ($N = 24$, $\rho_{\text{AL}} = 10^{-4}$), the error decreases an order of magnitude.

Table 2.4: *The (self-) errors in the proximity of two vesicles of RA* $= 0.99$ *with viscosity contrast* $\nu = 1$ *(top) and* $\nu = 10$ *(bottom) in the shear flow with and without the LRCA in Section 2.4 (see Fig. 2.12 and Fig. 2.13 for frames of the simulations at the coarsest resolution). The self-errors are computed with respect to the simulation in one row below. Also reported are the number of accepted and rejected time steps and the CPU time. The ground truth simulations of* $\nu = 1$ *and* $\nu = 10$ *take 16 and 36 hours, respectively. Both simulations have* $N = 96$ *and* $\Delta t =$ *5E-4.*

$\nu = 1$

		LRCA					Original			
N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{prox}}$	$\epsilon^{\text{s}}_{\text{prox}}$	Accepts	Rejects	Time (sec)	$\epsilon^{\text{g}}_{\text{prox}}$	Accepts	Rejects	Time (sec)
12	1E-2	5.3E-1	1.4E-1	110	2	57	5.5E-1	110	3	38
16	1E-3	3.4E-1	2.6E-1	373	3	199	1.1E-1	343	6	134
24	1E-4	6.5E-2	4.6E-2	1192	9	699	1.5E-2	1191	9	446
32	1E-5	1.8E-2		3766	13	2300	4.8E-3	3767	13	862
12	1E-2	5.3E-1	1.3E-1	110	2	57	5.5E-1	110	3	38
12	1E-3	2.8E-1	3.1E-2	333	3	193	3.8E-1	370	5	177
12	1E-4	4.3E-2		1187	6	805	3.1E-1	1592	19	599
16	1E-3	3.4E-1	3.5E-1	373	3	199	1.1E-1	343	6	134
16	1E-4	1.1E-2		1092	5	852	1.2E-1	1217	6	509

$\nu = 10$

		LRCA					Original			
N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{prox}}$	$\epsilon^{\text{s}}_{\text{prox}}$	Accepts	Rejects	Time (sec)	$\epsilon^{\text{g}}_{\text{prox}}$	Accepts	Rejects	Time (sec)
12	1E-2	6.8E-1	3.0E-1	99	3	52	3.8E-1	93	4	38
16	1E-3	3.0E-1	1.6E-1	318	7	181	7.5E-2	318	7	127
24	1E-4	1.2E-1	1.1E-1	1030	7	668	9.4E-3	1030	7	384
32	1E-5	1.1E-2		3160	8	2250	3.2E-3	3160	8	972
12	1E-2	6.8E-1	4.2E-1	99	3	52	3.8E-1	93	4	38
12	1E-3	1.8E-1	1.6E-1	328	3	272	2.8E-1	294	3	194
12	1E-4	2.4E-2		956	4	921	2.4E-1	958	4	563
16	1E-3	3.0E-1	2.0E-1	318	7	181	7.5E-2	318	7	127
16	1E-4	7.5E-2		955	4	1230	6.6E-2	955	4	580

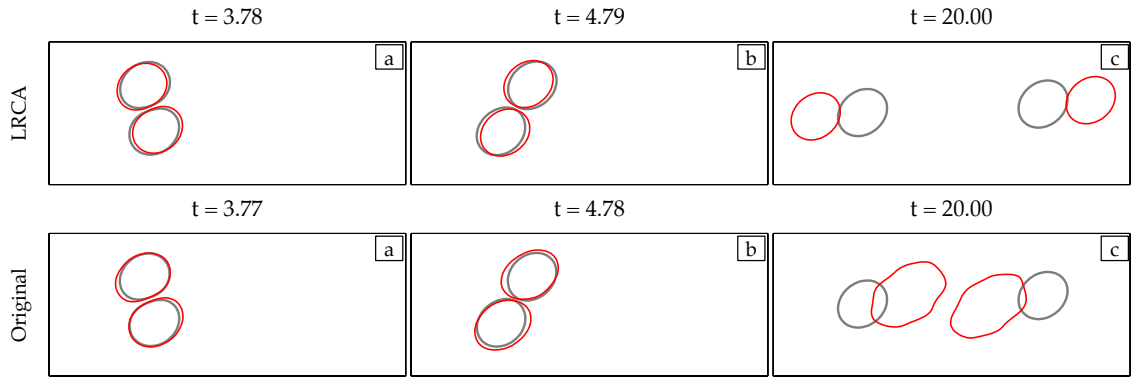


Figure 2.12: *Two vesicles with reduced area* 0.99*, viscosity contrast* $\nu = 1$*, and discretized with* $N = 12$ *points in shear flow. The error tolerance is* $\rho_{\mathrm{AL}} = 10^{-2}$*. In the top row, the LRCA are used, and in the bottom row, they are not. Similar to simulations shown in Fig. 2.10, although the errors in the proximity are very close with and without the LRCA at these resolutions (*$N = 12$ *and* $\rho_{\mathrm{AL}} = 1\mathrm{E}-2$*), the LRCA are necessary to maintain physical vesicle shapes.*

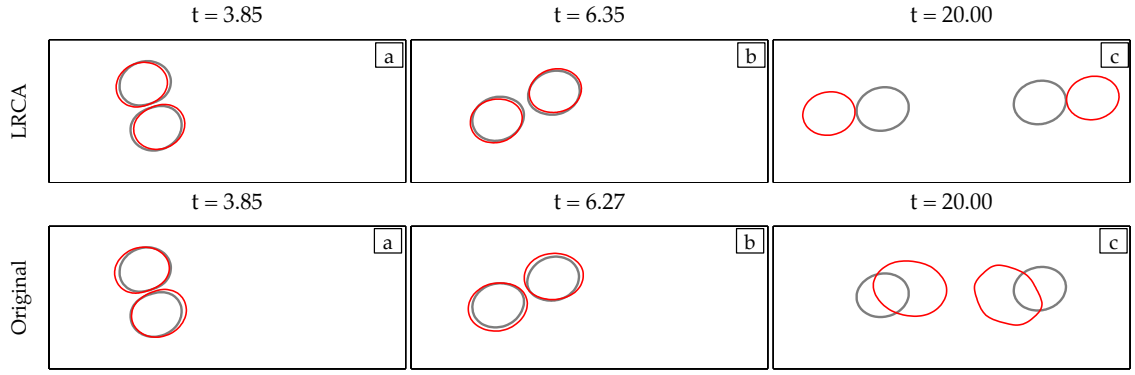


Figure 2.13: *Two vesicles with reduced area* 0.99*, viscosity contrast* $\nu = 10$*, and discretized with* $N = 12$ *points in shear flow. The error tolerance is* $\rho_{\mathrm{AL}} = 10^{-2}$*. In the top row, the LRCA are used, and in the bottom row, they are not.*

2.5.4 Stenosis Flow

Setup. We consider a single vesicle of reduced area 0.65 passing through a constricted tube (stenosis) without viscosity contrast (see Fig. 2.14). The flow is driven by a

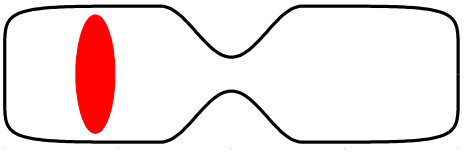


Figure 2.14: *Initial configuration of stenosis flow.*

Table 2.5: *Parameters of the ground truth for stenosis flow.*

Parameter	Value
Points on a vesicle N	128
Points on a wall N_{wall}	480
Number of SDC sweeps n_{sdc}	1
Time step size Δt	10^{-3}
CPU time	22 hours

parabolic flow profile at the intake and the outtake and the vesicle's initial height is 3.5 times larger than the size of the constriction. We choose a time horizon $T = 12$ so that the vesicle passes through the constriction. We simulate this example with $N = 12, 16, 24, 32$ points on the vesicle and $N_{\text{wall}} = 256$ points on the wall with and without the LRCA. The ground truth solution is formed with the parameters in Table 2.5.

Results. We again investigate the stability of simulations with and without the LRCA. In this example, reparameterization is necessary since the vesicle becomes highly deformed, which results in high frequencies in the shape that need to be removed. Time adaptivity and repulsion are necessary for the vesicle to pass through the constriction without crossing the outer boundary. In Fig. 2.15 the vesicle passing through the constriction at different resolutions with the LRCA are qualitatively compared with the ground truth. Even at the lowest resolution, the vesicle passes through the constriction,

Table 2.6: *The (self-) error in the center of the vesicle in a stenosis flow with and without the LRCA (see Fig. 2.15 for frames of these simulations using LRCA). The self-errors are computed with respect to the simulation in one row below. Also reported are the number of accepted and rejected time steps and the CPU time. The original simulations break when $N \leq 24$ because the vesicle-wall collisions cannot be handled. The ground truth simulation takes 22 hours with $N = 128$ and Δt = 1E-3.*

LRCA

N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{center}}$	$\epsilon^{\text{s}}_{\text{center}}$	Accepts	Rejects	Time (sec)
12	1E-1	1.7E-1	4.6E-2	29	6	83
16	1E-2	8.1E-2	3.2E-2	64	12	116
24	1E-3	2.6E-2	7.5E-3	208	32	348
32	1E-4	1.2E-2		567	27	887

Original

N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{center}}$	Accepts	Rejects	Time (sec)
12	1E-1	-	-	-	-
16	1E-2	-	-	-	-
24	1E-3	-	-	-	-
32	1E-4	1.4E-2	1312	155	1950

and the vesicle shape and center agree quite well with the ground truth. Whereas vesicle-wall collisions cannot be handled without the LRCA and the simulations break at these resolutions (i.e. $N \leq 24$).

We report the (self-) errors in the center, the number of accepted and rejected time steps, and the CPU time, both with and without the LRCA in Table 2.6. We see that without the algorithms, the low-resolution simulations are not stable with $N \leq 24$. At these resolutions, even with a very small time step, the dynamics when the shape is close to the solid wall can not be resolved. However, with the help of the LRCA, the simulations are stable and deliver acceptably accurate results in short CPU times. Even with $N = 32$ where the algorithms are unnecessary for stability, using them reduces the

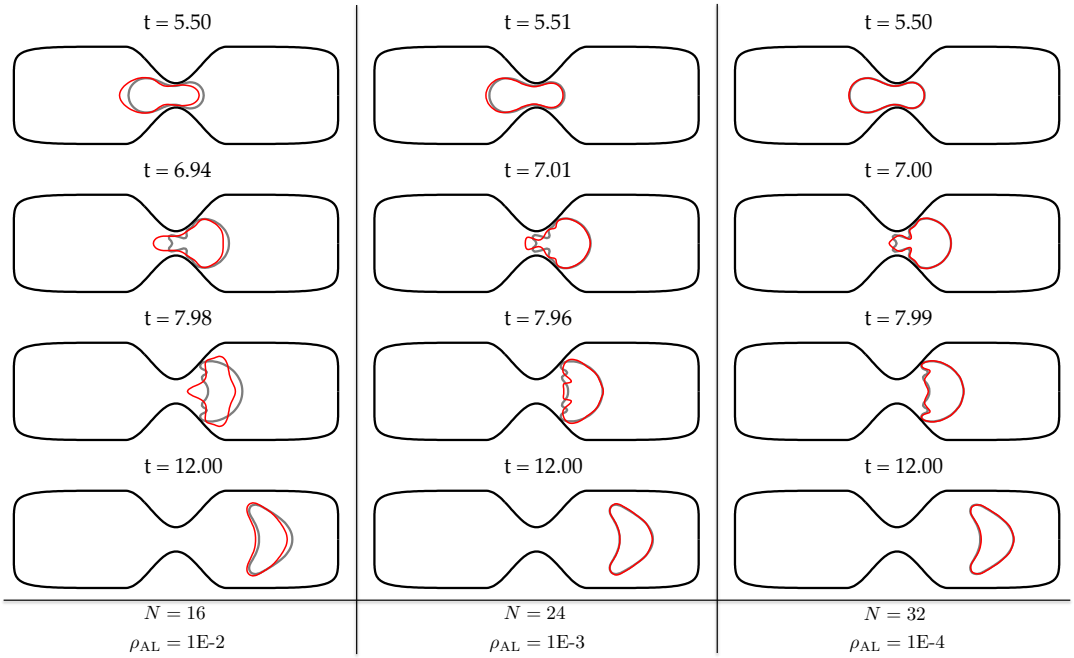


Figure 2.15: *A single vesicle passing through a constricted tube (***stenosis***). Here we vary the temporal and spatial resolutions simultaneously and use the LRCA. The spatial resolution and error tolerances are indicated at the bottom of each column. The wall is discretized with $N_{wall} = 256$ in all of the simulations. The gray vesicle is the ground truth. While these low-resolution simulations are stable and accurate with the LRCA, vesicle-wall collisions cannot be handled in the original simulations.*

total number of time steps resulting in a computationally faster method. Additionally, the self-error in vesicle's center decreases as the resolution increases.

2.5.5 Taylor-Green Flow

Setup. We consider four large vesicles of reduced area 0.65 in the periodic cell $(0, \pi)^2$ with the background Taylor-Green flow $\mathbf{u} = (\sin x \cos y, -\cos x \sin y)$. The vesicles occupy about 55% of the periodic cell (see Fig. 2.16). We color each vesicle separately for tracking purpose. The time horizon is $T = 20$ and we perform simulations with viscos-

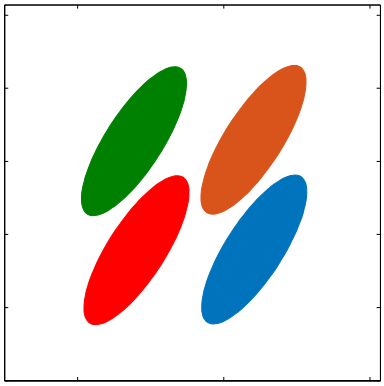


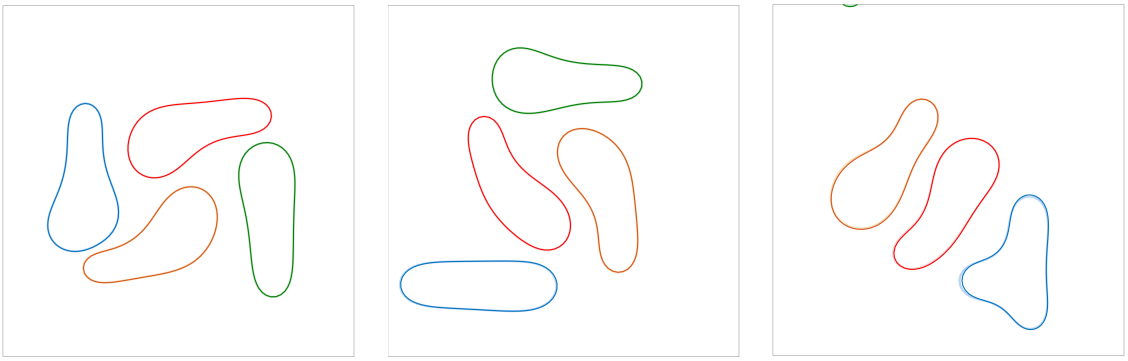
Figure 2.16: *Initial configuration of Taylor-Green flow.*

Table 2.7: *Parameters of the ground truth of Taylor-Green flow.*

Parameter	Value
Viscosity contrast ν	$\{1, 10\}$
Points per vesicle N $(\nu = 1)$	96
Points per vesicle N $(\nu = 10)$	64
Time step size Δt $(\nu = 1)$	2×10^{-4}
Time step size Δt $(\nu = 10)$	10^{-3}
Number of SDC sweeps n_{sdc}	1
CPU time $(\nu = 1)$	71.1 hours
CPU time $(\nu = 10)$	76.4 hours

ity contrasts $\nu = 1$ and $\nu = 10$. We simulate these examples with $N = 12, 16, 24, 32, 48$ points per vesicle, the error tolerances $\rho_{\text{AL}} = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$, with and without the LRCA. A ground truth solution for these examples is formed with the parameters in Table 2.7. We also demonstrate the convergence of the ground truth solution for the example with no viscosity contrast in Fig. 2.17.

Results. This example is more complex than the previous examples since there are interactions between multiple vesicles. Therefore, we expect that the LRCA are essential



(a) $t = 10$ (b) $t = 15$ (c) $t = 20$

Figure 2.17: *Convergence of the ground truth solution for the Taylor-Green flow with no viscosity contrast. The ground truth solution for this example is formed with $N = 96$ points per vesicle and a constant time step size of Δt = 2E-4, and shown with faded colors. We also superimpose vesicles from the solution with a lower spatial and temporal resolution ($N = 64$ and Δt = 1E-3, shown with bright colors). The errors in center and in inclination angle are ϵ_{center} = 1E-2 and ϵ_{IA} = 2E-2, respectively in the simulation with $N = 64$. Also see Table 2.8 for the details of the simulation with $N = 64$.*

for the stability at low resolutions. We summarize the results of the vesicles with $\nu = 1$ in Table 2.8 with and without the LRCA. We report the errors in the vesicles' centers, inclination angles, and effective viscosity, as well as the number of accepted and rejected time steps, and the total CPU time. The self-error is measured in terms of the effective viscosity only. Also, in Fig. 2.18 we show snapshots from simulations at four different resolutions along with the ground truth solution. We see that the LRCA result in stability at much lower resolutions, but the errors in the center and inclination angle of the vesicles are large (i.e., $\mathcal{O}(1)$). The reason for that is this example has more vesicle-vesicle interactions than the previous two and the near collisions lead to more chaotic flows [7, 123]. Convergence in terms of the local error measures such as the error in center and inclination angle requires fine resolutions (i.e., at

Table 2.8: *The maximum errors in the vesicles' centers, inclination angles, and the effective viscosity of four vesicles in the Taylor-Green flow with no viscosity contrast and with the LRCA. The ground truth simulation takes 71.1 hours with $N = 96$ and $\Delta t = 2E\text{-}4$.*

LRCA

N	ρ_{AL}	$\epsilon^g_{\text{center}}$	ϵ^g_{IA}	$\epsilon^g_{\mu_{\text{eff}}}$	$\epsilon^s_{\mu_{\text{eff}}}$	Accepts	Rejects	Time (sec)
12	1E-1	1.0E+0	9.7E-1	4.1E-1	3.7E-1	35	7	50
16	1E-2	1.8E+0	2.0E+0	5.0E-2	6.5E-3	92	8	106
24	1E-3	1.7E+0	2.1E+0	5.6E-2	2.3E-2	326	13	419
32	1E-4	1.6E+0	2.0E+0	3.4E-2	5.0E-3	1080	15	1390
48	1E-5	1.5E+0	4.2E-1	3.0E-2	4.5E-3	3437	26	5990
64	1E-3	1.0E-2	2.0E-2	1.2E-3		20001	-	61200
12	1E-1	1.0E+0	9.7E-1	4.1E-1	3.3E-1	35	7	50
12	1E-2	7.4E-1	2.0E+0	1.1E-1	9.9E-2	104	15	143
12	1E-3	8.3E-1	2.0E+0	2.3E-2	9.1E-3	312	17	486
12	1E-4	5.1E-1	1.7E+0	1.4E-2		1458	28	2200
16	1E-2	1.8E+0	2.0E+0	5.0E-2	3.9E-2	92	8	106
16	1E-3	1.7E+0	2.0E+0	2.5E-2	1.4E-2	333	18	359
16	1E-4	1.6E+0	1.8E+0	1.2E-2		1135	18	1610

Original

N	ρ_{AL}	$\epsilon^g_{\text{center}}$	ϵ^g_{IA}	$\epsilon^g_{\mu_{\text{eff}}}$	$\epsilon^s_{\mu_{\text{eff}}}$	Accepts	Rejects	Time (sec)
12	1E-1	-	-	-	-	-	-	-
16	1E-2	-	-	-	-	-	-	-
24	1E-3	-	-	-	-	-	-	-
32	1E-4	1.5E+0	1.9E+0	2.2E-1	2.1E-1	1064	16	736
48	1E-5	5.4E-1	2.8E-1	1.5E-2		3306	29	3480
12	1E-1	-	-	-	-	-	-	-
12	1E-2	-	-	-	-	-	-	-
12	1E-3	-	-	-	-	-	-	-
12	1E-4	-	-	-	-	-	-	-
16	1E-2	-	-	-	-	-	-	-
16	1E-3	-	-	-	-	-	-	-
16	1E-4	-	-	-	-	-	-	-

Figure 2.18: *Four vesicles in the Taylor-Green flow with no viscosity contrast and with the LRCA. The spatial resolutions and the error tolerances are given at the bottom of each column. Faded vesicles correspond to the ground truth and the low-resolution counterparts are in bright colors.*

least $N = 64$ for the no viscosity contrast case). Fig. 2.18 shows that the centers and inclination angles of vesicles in the low-resolution simulations are close to those of the ground truth over a short time. As the vesicles interact more, the errors accumulate

Table 2.9: *The maximum errors in the vesicles' centers, inclination angles, and the effective viscosity of four vesicles in a Taylor-Green flow with viscosity contrast* 10 *and with the LRCA. The ground truth simulation takes 76.4 hours with* $N = 64$ *and* $\Delta t = 1E\text{-}3$.

LRCA

N	ρ_{AL}	$\epsilon^g_{\text{center}}$	ϵ^g_{IA}	$\epsilon^g_{\mu_{\text{eff}}}$	$\epsilon^s_{\mu_{\text{eff}}}$	Accepts	Rejects	Time (sec)
12	1E-1	2.9E+0	2.0E+0	3.4E-1	2.4E-1	36	9	50
16	1E-2	1.6E+0	2.0E+0	1.3E-1	8.9E-2	91	7	87
24	1E-3	1.6E+0	5.9E-1	4.7E-2	9.2E-3	282	10	371
32	1E-4	3.4E-1	1.2E-1	3.8E-2	7.2E-2	881	10	1210
48	1E-5	1.1E-1	5.4E-2	3.6E-2		2835	15	8650
12	1E-1	2.9E+0	2.0E+0	3.4E-1	2.5E-1	36	9	50
12	1E-2	1.7E+0	2.0E+0	1.2E-1	1.9E-2	97	8	150
12	1E-3	1.3E+0	2.0E+0	1.0E-1	2.3E-2	284	10	595
12	1E-4	1.1E-1	1.8E+0	8.1E-2		888	16	2090
16	1E-2	1.6E+0	2.0E+0	1.3E-1	8.7E-2	91	7	87
16	1E-3	1.6E+0	2.0E+0	5.0E-2	7.8E-2	286	11	415
16	1E-4	2.5E-1	2.0E+0	3.1E-2		899	11	1670

Original

N	ρ_{AL}	$\epsilon^g_{\text{center}}$	ϵ^g_{IA}	$\epsilon^g_{\mu_{\text{eff}}}$	$\epsilon^s_{\mu_{\text{eff}}}$	Accepts	Rejects	Time (sec)
12	1E-1	-	-	-	-	-	-	-
16	1E-2	-	-	-	-	-	-	-
24	1E-3	-	-	-	-	-	-	-
32	1E-4	3.8E-1	1.7E-1	2.7E-2	5.0E-2	894	11	940
48	1E-5	9.7E-2	5.8E-2	2.5E-2		2786	12	6380
12	1E-1	-	-	-	-	-	-	-
12	1E-2	-	-	-	-	-	-	-
12	1E-3	-	-	-	-	-	-	-
12	1E-4	-	-	-	-	-	-	-
16	1E-2	-	-	-	-	-	-	-
16	1E-3	-	-	-	-	-	-	-
16	1E-4	-	-	-	-	-	-	-

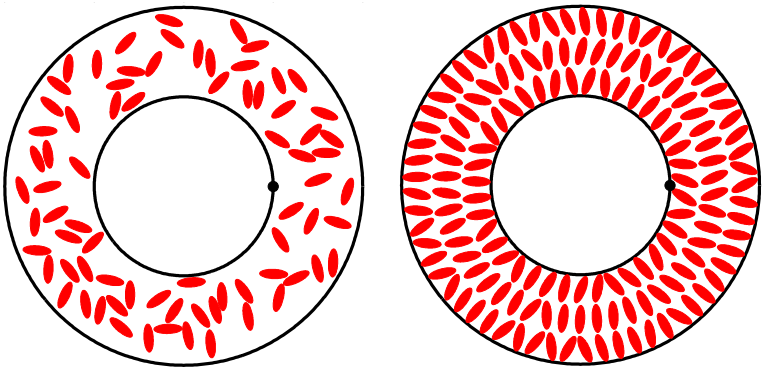
and result in diverging long-term behavior of an individual vesicle. However, the error of the effective viscosity is satisfactory. In contrast, without the LRCA, stability is not

achieved until $N = 32$ since vesicle-vesicle collisions cannot be handled accurately. Smaller errors can be achieved without the LRCA, but this requires a resolution of $N = 48$. At the two lowest resolutions, we increase the temporal resolution without changing the spatial resolution. While using the LRCA the errors decrease further with increasing temporal resolutions and the CPU times are still shorter than those with higher spatial resolutions, these simulations are not stable without the LRCA. The error in the effective viscosity with respect to the ground truth and the self-error decrease as the resolution is increased except when the resolution increases from ($N = 16$, $\rho_{\text{AL}} = 10^{-2}$) to ($N = 24$, $\rho_{\text{AL}} = 10^{-3}$). However, increasing the resolution further leads the (self-) error to decrease.

We repeat these experiments with viscosity contrast $\nu = 10$ and we report the results in Tables 2.9. Again, we see that with the LRCA, the errors in the center and inclination angle are large and the error in the viscosity contrast is small. Without the LRCA, stability requires $N = 32$ points, and smaller errors than than the those with our algorithms requires $N = 48$ points. Here, for the resolution ($N \geq 16$, $\rho_{\text{AL}} \geq 10^{-2}$) the (self-) error in the effective viscosity is in the order of $\mathcal{O}(10^{-2})$ so we observe the (self-) convergence. Although the error in the effective viscosity with respect to the ground truth always monotonously decreases, the self-error increases as the resolution increases from ($N = 24$, $\rho_{\text{AL}} = 10^{-3}$) to ($N = 32$, $\rho_{\text{AL}} = 10^{-4}$). The reason is that the effective viscosities for ($N = 32$, $\rho_{\text{AL}} = 10^{-4}$) and ($N = 48$, $\rho_{\text{AL}} = 10^{-5}$) are equally far from the ground truth but one is greater than the ground truth while the other is less. Therefore, the difference between them is higher than the difference between each of them and the ground truth. That is why the errors with respect to the ground truth

are close while the self-error is high.

2.5.6 Couette Flow



(a) Suspension at $\phi = 20\%$ (b) Suspension at $\phi = 40\%$

Figure 2.19: *Initial configuration of vesicles in Couette flows for area fractions $\phi = 20\%$ and $\phi = 40\%$. For the ground truth solutions, we use only the local area-length correction algorithm among the LRCA.*

Table 2.10: *Parameters of the ground truth for Couette flow examples.*

Parameter	Value
Points per vesicle N	96
Points per wall N_{wall}	256
Number of SDC sweeps n_{sdc}	1
Time step size Δt	10^{-2}
CPU time($\phi = 20\%$)	3 weeks
CPU time($\phi = 40\%$)	1 month

Setup. We consider two Couette flows with area fractions $\phi = 20\%$ (75 vesicles) and $\phi = 40\%$ (150 vesicles) without viscosity contrast (see Fig. 2.19). The inner boundary has radii $R_1 = 10$ and is rotating with constant angular velocity while the

outer boundary has radii $R_2 = 20$ and is stationary. We choose a time horizon $T = 100$ which results in the inner cylinder completing approximately 16 rotations. We simulate these cases with $N = 16, 24, 32, 48$ points per vesicle, $N_{\text{wall}} = 128$ points per wall, error tolerances $\rho_{\text{AL}} =$ 1E-2, 2E-3, 1E-3, 5E-4, 2E-4 and the LRCA. The ground truth solution for these examples use the parameters in the caption of Fig. 2.19.

Results. A Couette apparatus is often used to investigate properties of suspensions such as shear-induced diffusion [132] and effective viscosity. High volume fraction suspensions are of particular importance since red blood cells make up approximately 45% of human blood [52]. In addition, long time horizons are required for statistical analysis. Therefore, there are a large number of interactions between vesicles and walls, near collisions, and highly deformed vesicles. The interactions and shapes can be resolved with fine resolutions, but at a significant computational cost.

We are interested in the errors of upscaled variables. We report the (self-) errors in effective viscosity, space-time average and time average of the L^2 norm of a velocity field in Table 2.11. We also present the frames from the simulations of the suspension at $\phi = 20\%$ in Fig. 2.20 and the suspension at $\phi = 40\%$ in Fig. 2.21 at various resolutions. As in the previous example, the vesicle trajectories are not captured by the simulations with the LRCA. However, the errors in the upscaled quantities are at an acceptable as they are in the order of $\mathcal{O}(10^{-2})$ even with $N = 16$ points per vesicle. In addition, the computation speed-up is significant; the low-resolution runs required no more than a little over a day ($\phi = 20\%$) and less than a week ($\phi = 40\%$). In contrast, the ground truth simulations required 3 weeks ($\phi = 20\%$) and a month ($\phi = 40\%$).

Table 2.11 shows that the errors in the effective viscosity, space-time average of velocity and the time average of the L^2 norm of the velocity with respect to the ground truth are already $\mathcal{O}(10^{-2})$ and decrease as the resolution increases. The self-errors in these quantities are also $\mathcal{O}(10^{-2})$, however it oscillates as the resolution increases. The small self-error indicates that the convergence to the ground truth is attained. The reason of the oscillation is that there is no significant change in the error with respect to the ground truth as the resolution increases, these quantities oscillate around the ground truth. That is why although the error with respect to the ground truth is small, the error with respect to the simulation with a higher resolution is large.

Table 2.11: *The (self-) errors in the effective viscosity $\epsilon_{\mu_{\text{eff}}}$, time-space average of the velocity $\epsilon_{\langle v \rangle}$, and the time average of the L^2 norm of the velocity $\epsilon_{\langle V \rangle}$ of suspensions at $\phi = 20\%$ (Fig. 2.20) and $\phi = 40\%$ (Fig. 2.21) in the Couette flow with the LRCA . The self-errors are computed with respect to the simulation in one row below. Also reported are the number of accepted and rejected time steps and the CPU time. The ground truth simulations use $N = 96$ and $\Delta t =$ 1E-2 and take 3 weeks for $\phi = 20\%$ and a month for $\phi = 40\%$.*

$\phi = 20\%$

N	ρ_{AL}	$\epsilon^g_{\mu_{\text{eff}}}$	$\epsilon^s_{\mu_{\text{eff}}}$	$\epsilon^g_{\langle v \rangle}$	$\epsilon^s_{\langle v \rangle}$	$\epsilon^g_{\langle V \rangle}$	$\epsilon^s_{\langle V \rangle}$	Accepts	Rejects	Time (hours)
16	2E-2	5.5E-2	1.8E-2	7.9E-2	4.4E-2	2.3E-2	4.7E-3	507	74	9.3
24	1E-3	3.8E-2	4.7E-3	3.6E-2	1.3E-2	1.8E-2	5.7E-3	2160	55	32.2
32	5E-4	3.3E-2	1.1E-2	2.3E-2	6.6E-3	1.2E-2	2.5E-3	3175	23	52
48	2E-4	2.2E-2		1.7E-2		9.8E-3		5055	67	100.3
16	2E-2	5.5E-2	2.2E-2	7.9E-2	4.1E-2	2.3E-2	5.6E-3	507	74	9.3
16	1E-3	3.4E-2		4.0E-2		1.7E-2		2499	82	32.8
24	2E-2	4.2E-2	3.9E-3	8.8E-2	5.3E-2	2.9E-2	1.1E-2	402	31	10.1
24	1E-3	3.8E-2		3.6E-2		1.8E-2		2160	55	32.2

$\phi = 40\%$

N	ρ_{AL}	$\epsilon^g_{\mu_{\text{eff}}}$	$\epsilon^s_{\mu_{\text{eff}}}$	$\epsilon^g_{\langle v \rangle}$	$\epsilon^s_{\langle v \rangle}$	$\epsilon^g_{\langle V \rangle}$	$\epsilon^s_{\langle V \rangle}$	Accepts	Rejects	Time (hours)
16	1E-2	2.7E-2	2.3E-2	1.3E-1	5.7E-2	5.9E-2	1.2E-2	1007	125	62.8
24	1E-3	3.1E-3	4.4E-2	7.8E-2	3.5E-2	4.7E-2	1.7E-2	2894	67	138.6
32	5E-4	3.9E-2	1.9E-2	4.4E-2	3.6E-2	3.0E-2	2.5E-2	4257	42	197.8
48	2E-4	2.2E-2		7.9E-3		5.7E-3		6384	89	354.4
16	1E-2	2.7E-2	4.3E-2	1.3E-1	7.3E-2	5.9E-2	2.9E-2	1007	125	62.8
16	1E-3	1.6E-2		6.1E-2		3.1E-2		3517	110	156.1
24	1E-2	1.9E-2	1.6E-2	1.5E-1	7.7E-2	7.2E-2	2.6E-2	806	37	42.2
24	1E-3	3.1E-3		7.8E-2		4.7E-2		2894	67	138.6

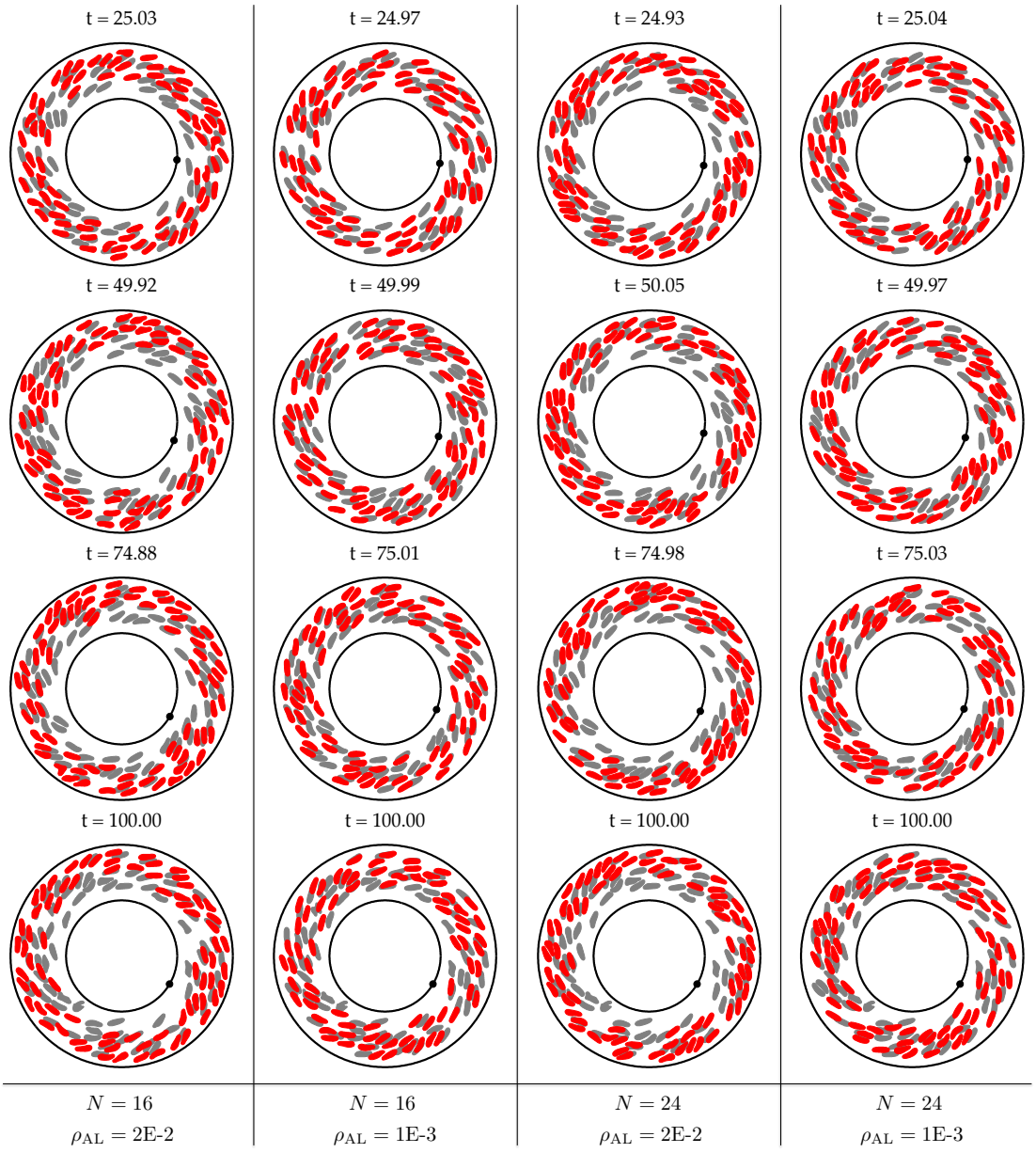


Figure 2.20: *75 vesicles in Couette flow (area fraction $\phi = 20\%$). The ground truth is shown with gray vesicles superimposed with the low-resolution counterpart (red vesicles). The simulations are compared at different, but comparable times due to the use of adaptive time stepping.*

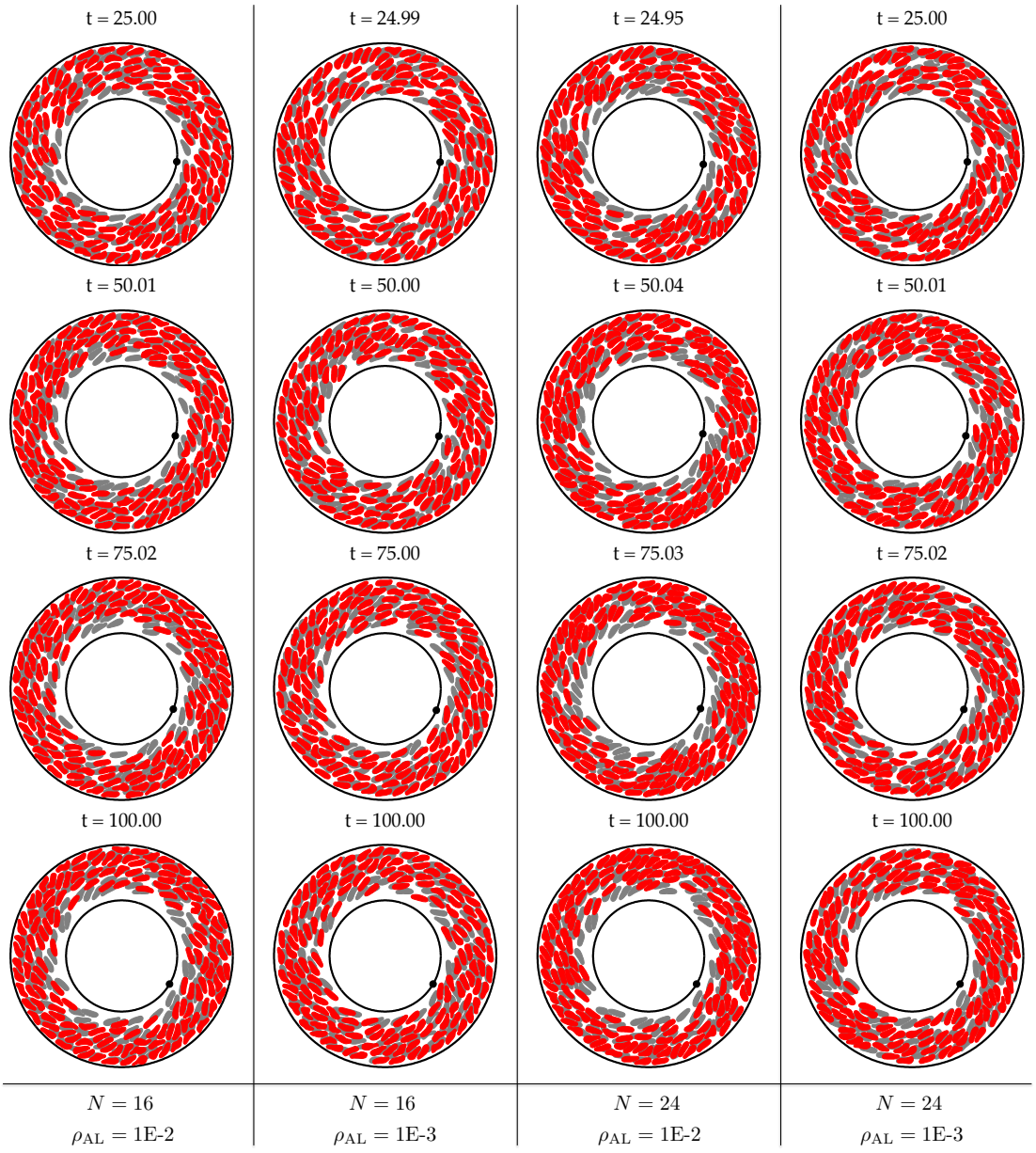
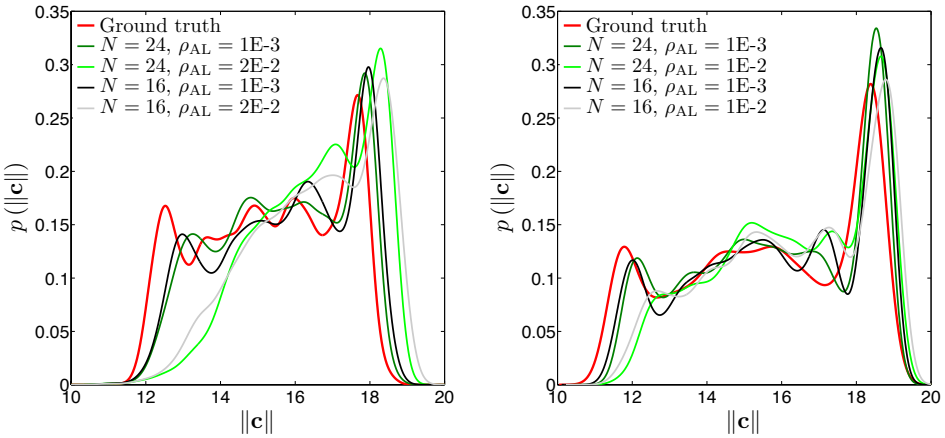


Figure 2.21: *150 vesicles in Couette flow (area fraction $\phi = 40\%$). The ground truth is shown with gray vesicles superimposed with the low-resolution counterpart (red vesicles). The simulations are compared at different, but comparable times due to the use of adaptive time stepping.*

Cell-Free Layer. Next, we investigate how accurately the low-resolution simulations can capture the statistics of the vesicle locations. In this setup, vesicles are known to migrate away from the walls resulting in a so-called cell-free layer near the walls [101]. This layer is captured by our coarse spatial but fine temporal resolution simulations, i.e. low error tolerances ρ_{AL} (see Figures 2.20 and 2.21). However, at the high error tolerances the cell-free layer is thicker than the ground truth (first and third columns). To further demonstrate this point, we plot the probability distribution functions of distances of the vesicles' centers to the origin throughout the simulations in Fig. 2.22. The figure shows that the simulations with the error tolerance $\rho_{\mathrm{AL}} = 10^{-3}$ estimate the cell-free layer accurately at both spatial resolutions, while with tolerance $\rho_{\mathrm{AL}} = 10^{-2}$, the cell-free layer is thicker than the ground truth. This suggests that although the local errors are too large in the simulations of dense suspensions at low resolutions, the upscaled quantities and statistics are rather insensitive to the local errors and can be accurately captured by the low-resolution simulations.

Statistics of the Velocity Field. We also use simulations of vesicle suspensions in a Couette apparatus to infer mixing properties of the suspensions [79]. For this reason, it is important to estimate the velocity field accurately. We compute the error in the space-time averages of the velocity field discussed above. In Fig. 2.23 we present statistics of the magnitude of the velocity field, $\|\mathbf{V}\|$, at points equally distributed in the azimuthal direction at three different radii ($\frac{r-R_1}{R_2-R_1} = 0.2, 0.5, 0.8$). Then we plot the probability distribution function of $\|\mathbf{V}\|$ in Fig. 2.23. In the absence of vesicles, $\mathbf{V}$ is only a function of the radial position in a Couette flow. However, the presence of vesicles



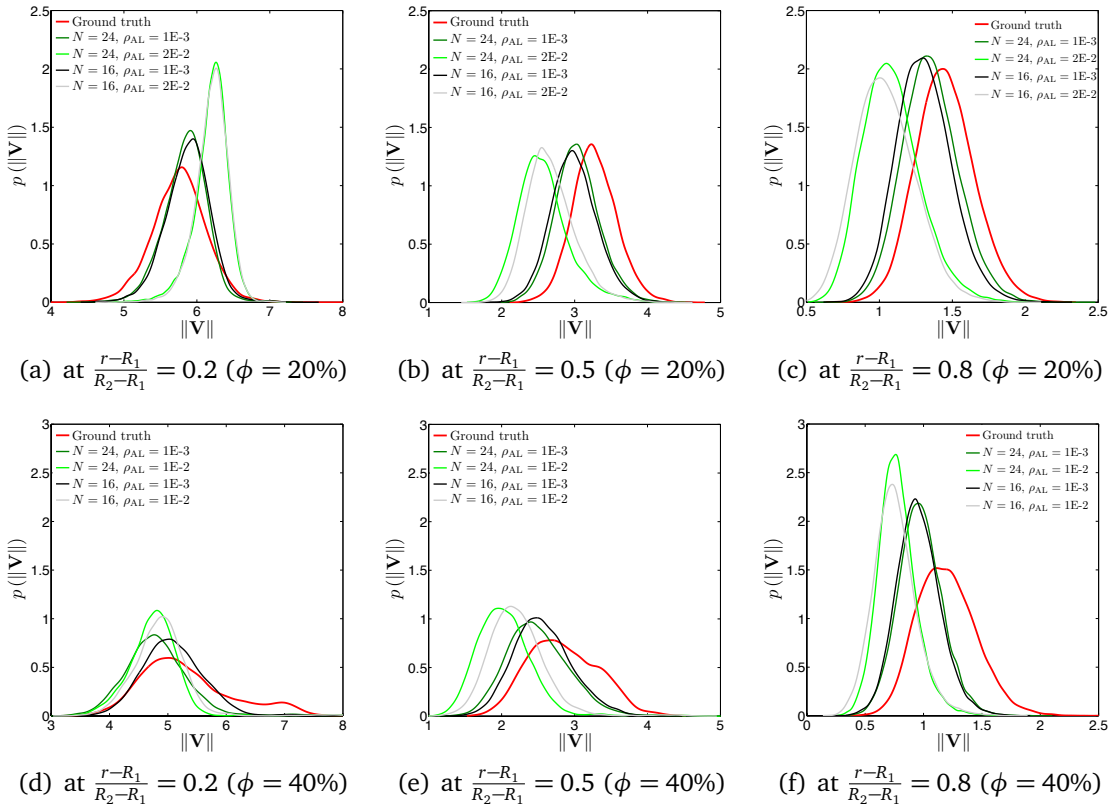
(a) A Couette flow with $\phi = 20\%$ (b) A Couette flow with $\phi = 40\%$

Figure 2.22: *Statistics of the vesicles' concentration in a Couette flow. We plot the probability distributions of distances of the vesicles' centers to the origin for the suspensions with area fractions* $\phi = 20\%$ *(left) and* $\phi = 40\%$ *(right).*

perturbs the velocity field. The low-resolution simulations with the error tolerances $\rho_{\mathrm{AL}} = 10^{-3}$ approximate the statistics of the velocity field closely. Similar to the statistics to capture the cell-free layer (Fig. 2.22), higher temporal resolutions provide more accurate velocity statistics while the spatial resolution does not significantly affect the results (see Fig. 2.23).

2.5.7 Microfluidic Device

Deterministic lateral displacement (DLD) is a microfluidic technique to separate particles depending on their sizes and deformability without using any external force [71]. A DLD device consists of matrix of pillars, where the rows are arranged at an angle with the x-axis (horizontal) and the imposed velocity profile (or pressure difference) is aligned with the x-axis. When a particle (e.g., rigid particles, vesicles,

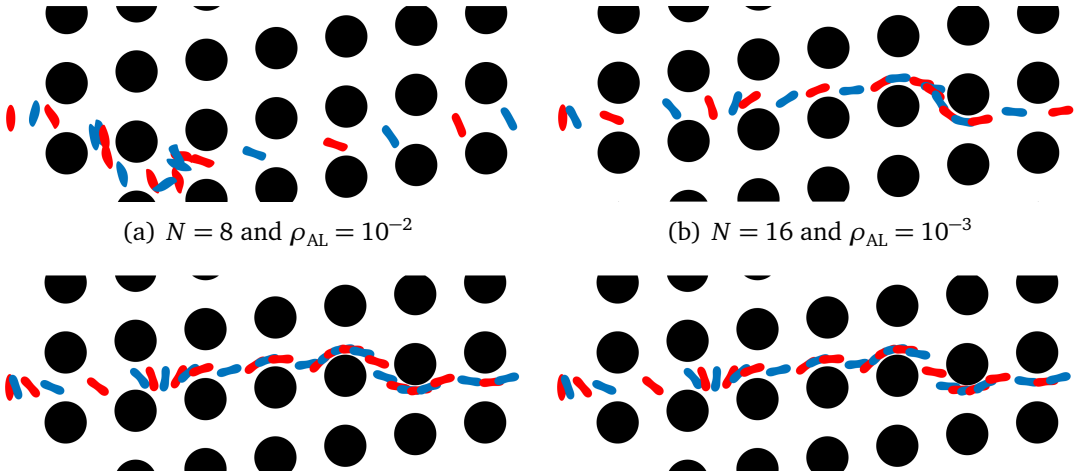


(a) at $\frac{r-R_1}{R_2-R_1} = 0.2$ ($\phi = 20\%$) (b) at $\frac{r-R_1}{R_2-R_1} = 0.5$ ($\phi = 20\%$) (c) at $\frac{r-R_1}{R_2-R_1} = 0.8$ ($\phi = 20\%$)

(d) at $\frac{r-R_1}{R_2-R_1} = 0.2$ ($\phi = 40\%$) (e) at $\frac{r-R_1}{R_2-R_1} = 0.5$ ($\phi = 40\%$) (f) at $\frac{r-R_1}{R_2-R_1} = 0.8$ ($\phi = 40\%$)

Figure 2.23: *Statistics of the velocity field in the Couette flow. We compute the probability distribution function of the velocity magnitudes at points distributed equally in the azimuthal direction at three different radii. The top plots correspond to the area fraction $\phi = 20\%$ and the bottom plots correspond to the area fraction $\phi = 40\%$.*

or red blood cells) enters the device it typically exhibits two modes of motion. Either it *"displaces"* or it *"zig-zags"*. These two terms are explained in Fig. 2.24. The basic idea is that if we want to separate particles, we design a DLD device in which one set of particles displaces and the other zig-zags. The experimental study [19] shows that the technique can be used to separate red blood cells depending on their deformability. Follow up numerical studies [97, 145, 166, 175, 179] systematically analyzed



(a) $N = 8$ and $\rho_{\mathrm{AL}} = 10^{-2}$ (b) $N = 16$ and $\rho_{\mathrm{AL}} = 10^{-3}$

(c) $N = 32$ and $\rho_{\mathrm{AL}} = 10^{-4}$ (d) $N = 64$ and $\rho_{\mathrm{AL}} = 10^{-4}$

Figure 2.24: *Snapshots of zig-zagging RBCs from our low-resolution (the first three) and the ground truth (at the bottom) simulations of the microfluidic device. The regular alternation between blue and red RBCs represents sequential frames with variable time intervals. The device uses the technique called deterministic lateral displacement (DLD) to separate cells based on their deformability. Our DLD device consists of arrays of circular pillars (shown in black) and an exterior wall (not shown). The suspension flows from left to right (aligned with the horizontal axis). We impose a parabolic velocity at the intake and the outtake that causes a healthy red blood cell to cross the inclined rows of pillars. This crossing is called* "zig-zagging" *and has also been observed experimentally [19]. The ground truth (d) shows just before the last two columns of pillars, the cell goes around the pillar and crosses rows, thus, it "zig-zags". If a cell does not zig-zag, we say that the cell displaces (laterally) along a row of pillars.*

the separation of red blood cells using DLD and successfully reproduced the experimental results. Among these numerical studies [145, 175, 179] are two-dimensional and [97, 166] are three-dimensional. Here, we want our 2D model to reproduce these numerical and experimental results using as coarse discretization as possible.

Setup. The DLD device we consider here consists of circular pillars with a diameter of 15 μm bounded by an exterior wall (not shown). We impose a Poiseuille flow as

a velocity boundary condition at the intake and the outtake, and hence the velocity between two laterally adjacent pillars is parabolic. We consider a healthy red blood cell which has a reduced area of 0.65 and a viscosity contrast $\nu = 10$. The lengths of the long and short axes of the RBC are $8\,\mu m$ and $3\,\mu m$. The inclination angle of the device is 0.17 rad and the center-to-center distance between the pillars is 25 μm. The setup of this DLD device (geometry and imposed velocity) are such so that the cell zig-zags (see Fig. 2.24).

We discretize the exterior wall with $N_{\text{wall}} = 3712$ points, the pillars with $N_{\text{pillar}} = 64$ points. In our convergence study we do not change these resolutions. The repulsion length scale we use here is $d_{\text{min}} = 0.5h_{\text{max}}$. We start with $N = 8$ points per vesicle and $\rho_{\text{AL}} = 10^{-2}$. If the simulation can be completed within the allocated CPU time T_{CPU}, we perform a self-convergence test to determine the accuracy of the low-resolution solution. For this purpose we run another simulation of the example with a higher resolution $N = 16$ and $\rho_{\text{AL}} = 10^{-3}$. This *"ground truth"* solution is performed using $N = 64$ points per vesicle and $\rho_{\text{AL}} = 10^{-4}$. For reference, the ground truth simulation Fig. 2.24(d) requires 5.6 hours (on a single workstation) and is in agreement with the experimental results reported in [19].

Results. We are interested in capturing the true motion of the cells, i.e., displacement vs zig-zag and the correct point of zig-zagging so we can properly characterize the behavior of the device. We report a qualitative error metric (zig-zagging or not, and the pillar in which zig-zagging takes place). We also report two quantitative errors, one highly sensitive to the accuracy of the calculation and one less sensitive one. The

first one (sensitive) is the error in the vesicle's center ϵ_{center}, specifically, its maximum over all time steps in Table 2.12. The second error (less sensitive to numerical errors) is in the time it takes for the RBC to travel to the end of the device. We denote this error measure by ϵ_{T}. In terms of computational efficiency, we also report the number of accepted and rejected time steps, and the total CPU time.

Our black-box solver took us to an accurate solution as follows: the simulation with $N = 8$ and $\rho_{\text{AL}} = 10^{-2}$ was completed within 3 hours. Then in order to estimate its accuracy we performed another simulation with $N = 16$ and $\rho_{\text{AL}} = 10^{-3}$, which took slightly longer than 2 hours. The self-error of the first simulation in the vesicle's center turned out to be $\epsilon^{\text{s}}_{\text{center}} = 2.8$, which is a close estimate of the error in the vesicle's center compared to the ground truth $\epsilon^{\text{g}}_{\text{center}} = 3.2$ and not acceptable. The self-error in the travel time is also large, i.e. $\epsilon^{\text{s}}_{\text{T}} = 0.25$. So another simulation with a higher resolution ($N = 24$, $\rho_{\text{AL}} = 10^{-4}$) was performed to measure the accuracy of the second simulation with $N = 16$ and $\rho_{\text{AL}} = 10^{-3}$. The self-error in the center still remains large but the self-error in the travel time decreases to $\mathcal{O}(10^{-2})$. Since this flow has several vesicle-wall interactions, the error in the center might be large at the low resolutions as in the Taylor-Green flow and the Couette apparatus examples. Therefore, if the quantity of interest is the travel time or the pillar in which zig-zagging takes place, $N = 16$ and $\rho_{\text{AL}} = 10^{-3}$ seem to be sufficient for the accurate physics. We performed one more simulation with $N = 32$ and $\rho_{\text{AL}} = 10^{-4}$. This and the previous simulations had two times the CPU times of the first two runs. Additionally, the error in the vesicle's center or in the travel time did not improve further. So the self-convergence is achieved.

Table 2.12: *The (self-) errors in the vesicle's center ϵ_{center} and the vesicle's travel time to the end of the device ϵ_{T} for the simulations of the microfluidic device with the LRCA. The superscripts "s" and "g" indicate self-convergence errors and errors with the ground truth, respectively. The self-convergence errors are computed with respect to the simulation in one row below. Also reported are the number of accepted and rejected time steps and the CPU time. The ground truth simulation takes 5.6 hours with $N = 64$ and $\rho_{\text{AL}} = 10^{-4}$.*

N	ρ_{AL}	$\epsilon^{\text{g}}_{\text{center}}$	$\epsilon^{\text{s}}_{\text{center}}$	$\epsilon^{\text{g}}_{\text{T}}$	$\epsilon^{\text{s}}_{\text{T}}$	Accepts	Rejects	Time (hours)
8	1E-2	3.2E+0	2.8E+0	3.3E-1	2.5E-1	506	27	2.92
16	1E-3	1.0E+0	1.1E+0	1.0E-1	1.7E-2	396	20	2.06
24	1E-4	6.7E-1	2.4E-1	8.5E-2	3.1E-2	1227	40	5.22
32	1E-4	4.5E-1		5.6E-2		1228	40	5.25
8	1E-2	3.2E+0	2.5E+0	3.3E-1	1.3E-1	506	27	2.92
8	1E-4	3.3E+0		4.1E-1		2429	75	10.25
16	1E-3	1.0E+0	3.5E-1	1.0E-1	4.4E-2	396	20	2.06
16	1E-4	1.1E+0		1.4E-1		1195	45	5.14

In conclusion, the scheme correctly identifies the necessary resolution to resolve the quantities of interest. In this example $N = 16$ is sufficient to capture the correct zig-zagging behavior. All the simulations exhibit zig-zagging but the $N = 8$ case is completely off (see Fig. 2.24(a)). As we discuss, the simulation was run without changing any parameters, other than N and ρ_{AL}.

2.6 Conclusions

We have addressed issues with simulations of vesicle suspensions at low discretization resolutions. We have developed a robust method by introducing new schemes and implementing some standard techniques. An efficient scheme to determine an upsampling rate is used for computing the nonlinear terms without introducing spurious oscillations. A surface reparameterization algorithm smooths out vesicles' boundaries

by penalizing their high-frequency components. The area and arc- length of the vesicles are corrected at each time step to allow for long-time scale simulations without changing the governing equations. A new reliable adaptive time stepping scheme that works for all resolutions is used to choose the optimal time step size. Finally, a repulsion force between vesicles eliminates any chance of an non- physical collision. All these algorithms require certain parameters, which were set heuristically. So our solver can be used as a black-box. We show the capabilities of the solver in a real-world example of a microfluidic cell sorting technique. We have performed a systematic error analysis to investigate accuracy of low-resolution simulations. The presented low-resolution correction algorithms are essential for stable simulations. Furthermore, by using these algorithms we are able to accurately capture the statistics of the underlying flow accurately with a coarse discretization. One of the most impressive examples is the Couette flow. Its low-resolution simulation, which takes less than a week, estimates accurately the upscaled quantities such as effective viscosity and statistics computed by the high-fidelity simulation, which takes more than a month.

Chapter 3

Quantification of Mixing in Vesicle Suspensions Using Numerical Simulations in Two Dimensions

In this chapter[1] we present our study on mixing in vesicle suspensions in two dimensions. We simulate vesicle flows using our scheme introduced in the previous chapter and solve the advection-diffusion equation for the mixing of a solute using a pseudo-spectral scheme. We compare mixing in the suspension with mixing in a Couette apparatus without vesicles. On the one hand, the presence of vesicles in most cases, slightly suppresses mixing. This is because the solute can be only diffused across the vesicle interface and not advected. On the other hand, there exist spatial distributions of the solute for which the unperturbed Couette flow completely fails to mix whereas the presence of vesicles enables mixing. We derive a simple condition that relates the velocity and solute and can be used to characterize the cases in which the presence of vesicles promotes mixing.

3.1 Introduction

Vesicles play an important role in cellular transport. Artificial ones can be designed to target specific sites in a human body, such as tumors or diseased regions.

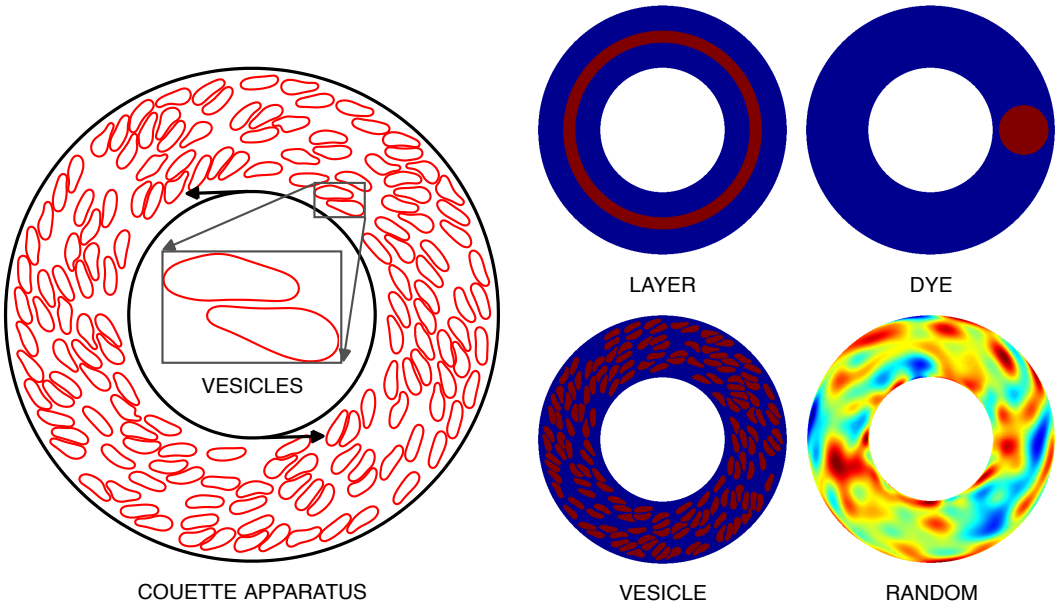
[1] This chapter is based on work that has been published in [79]. The authors equally contributed.

Targeted drug delivery enables applying drugs to a diseased site with a prescribed dose, which prevents damaging healthy parts of the body to some extent. There have been experimental studies to develop successful drug delivery systems [68, 157, 161]. These studies need to be complemented by numerical studies to help understand, control and optimize drug release. Here, we propose a numerical framework to characterize mixing in vesicle flows and investigate effects of several flow parameters on mixing. A recent study similar to ours [85] considered a polydisperse suspension, a liposome surrounded by red blood cells. The flow and the mass transfer in 2D are computed using the lattice Boltzmann method.

Methodology. We consider the advective and the diffusive mixing of a passive scalar in a Couette apparatus depicted in Fig. 3.1(a). The apparatus has a rotating inner cylinder and a stationary outer cylinder. The relevant dimensionless number of the transport problem is the ratio of the advective transport rate to the diffusive transport rate, or the Peclet number Pe,

$$\mathrm{Pe} = \overline{V} L_c / D. \tag{3.1}$$

Here $\overline{V}$ is the time average of the L^2 norm of the velocity field $\mathbf{v}$, i.e. $\overline{V} = \langle \|\mathbf{v}\|_{L^2} \rangle$ and measures the kinetic energy, L_c is the characteristic length scale (the diameter of the apparatus), and D is the diffusivity of the transported quantity. As an example, we discuss transport in microcirculation. The diameters of a capillary and an arteriole are $\mathcal{O}(10\ \mu\mathrm{m})$ and the mean velocities of blood flow in them are $\mathcal{O}(1)$ mm/s and $\mathcal{O}(10)$ mm/s, respectively [52, 133, 171]. The diffusivity of oxygen is $\mathcal{O}(10^{-3})$ mm^2/s [171]. Hence, the Peclet number for the transfer of oxygen ranges from 10 to 10^4.



(a) A vesicle suspension in a Couette apparatus (b) Initial conditions for mixing simulations

Figure 3.1: *Geometry and initial conditions used in Chapter 3. We study mixing in vesicle flows in a Couette apparatus (a) that consists of a rotating inner cylinder and a stationary outer cylinder. We, first, simulate vesicle flows to compute the velocity field perturbed by the presence of vesicles. This step is independent of the advection-diffusion equation since the transported quantity is passive. We, then, solve the transport problem for several initial conditions depicted in (b). Red colors correspond to high concentration (maximum value is 1) and blue colors correspond to low concentration (minimum value is 0).*

Vesicle suspensions have several parameters such as the distribution of sizes of the vesicles, the reduced area, the bending resistance, the area fraction (the ratio between the area occupied by the vesicles and the total area of the apparatus), and the viscosity contrast between the fluids in the interior and the exterior of a vesicle. All these parameters could affect mixing. Here, however, we consider only two main vesicle parameters, the area fraction and the viscosity contrast. Of course, another parameter is the imposed external velocity field. In our case it is the velocity generated

by the rotating inner cylinder in the Couette apparatus and we parameterize it by the Peclet number. Again, taking an example from microcirculation, the volume fraction of red blood cells in human blood is typically around 45% and their viscosity contrast with plasma ranges from 5 to 10 [52].

Our numerical simulations require two steps. First, we simulate the vesicle motion for various values of area fraction and viscosity contrast and then compute the velocity field on a Fourier-Chebyshev grid. Second, using this velocity field, we simulate mixing in the Couette apparatus for the initial conditions in Fig. 3.1(b) and denote the corresponding concentration of the solute with ϕ. We remark that numerical algorithms for the calculation of the velocity and ϕ are very different. The suspension dynamics are computed using the boundary integral equation method described in Section 2.3 while the advection-diffusion equation is solved using a pseudospectral method. We also simulate mixing of the same initial concentration in the apparatus with the same Peclet number but without any vesicles, and we denote this concentration by ϕ_0. Using ϕ_0, we define a mixing efficiency η as

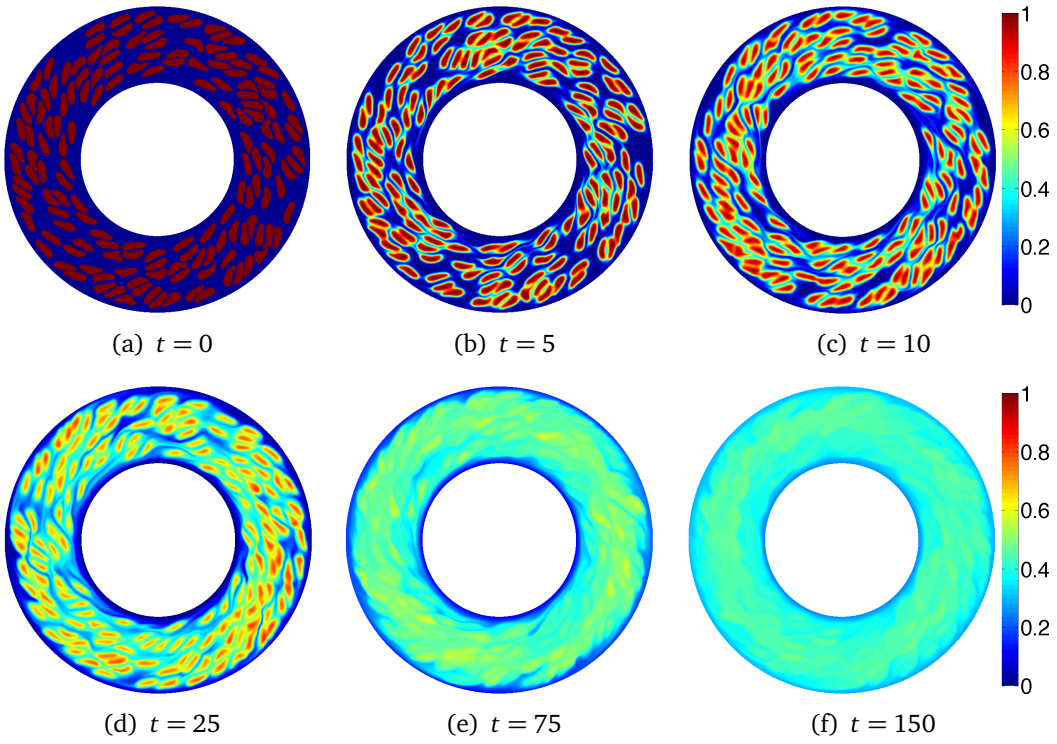
$$\eta = \frac{\|\phi_0\|}{\|\phi\|}, \tag{3.2}$$

which compares the mixing efficiency of the Couette apparatus with vesicles to that without vesicles (the default Couette flow). η greater than one means that the vesicle flow mixes better. When computing the Peclet number (3.1), we use the spatio-temporal average of the velocity field to quantify the advective transport rate. Since the velocity field $\mathbf{v}$ depends on the area fraction and the viscosity contrast, it changes with the area fraction and viscosity contrast of the suspension. Thus, we adjust the diffusivity D to keep the Peclet number the same in computing ϕ_0 and ϕ. In this manner, we

investigate the effects of area fraction and viscosity contrast on the mixing efficiency. Additionally, we look at these effects under various Peclet numbers for the initial conditions for the passively transported quantity (*the solute*) in Fig. 3.1(b). The membranes of the vesicles in our model are assumed to be impermeable for the background fluid (*the solvent*) and permeable for the solute.

Contributions. There is no unique way to define mixing. We discuss several of them and use a measure based on negative Sobolev norms of the concentration. One of the main findings of the study is since, in the model, Lagrangian trajectories do not cross the vesicle membrane, this has the effect of reducing advective mixing. Overall, we find that for the same average Peclet number, the presence of vesicles *slightly reduces mixing*. Interestingly, however, this is not always the case. There exist certain rather special initial conditions for the passively transported quantity that this is not the case. For these conditions in the absence of vesicles there is no advective mixing while the presence of vesicles *increases* mixing. One such initial condition is the *LAYER* initial condition in Fig. 3.1(b).

Limitations. The main limitation is that we only consider a specific two-dimensional flow. So, generalizations to other type of flows are not immediate. Also, we consider several metrics for mixing, but other metrics can be considered. Although we use a constant diffusion coefficient for the solute in the passive advection-diffusion model, the diffusion coefficient can depend on whether the diffusion is considered in the fluid bulk, the fluid enclosed by the vesicle, or the vesicle membrane. We also assume



(a) $t = 0$ (b) $t = 5$ (c) $t = 10$

(d) $t = 25$ (e) $t = 75$ (f) $t = 150$

Figure 3.2: *Snapshots from a mixing simulation in a vesicle suspension. The suspension has a* 40% *area fraction of vesicles with no viscosity contrast between the fluids in the interior and the exterior of vesicles. The initial condition for the solute is one inside the vesicles and zero outside the vesicles. This advection-dominated transport problem has Pe = 1E+4. The time horizon ($T_h = 150$) corresponds to 24 revolutions of the inner cylinder.*

that the vesicle membrane is impermeable to the suspending fluid. Additionally, we investigate the effects of only the area fraction, the viscosity contrast, and the Peclet number on mixing.

Related work. Mixing has been studied extensively as it is important in many scientific and industrial settings. Classical works in mixing [9, 102] consider large scale systems such as combustion in engines and pollution in seas. Mixing is of fundamen-

tal importance to many biological systems. The survival of many microorganisms requires transport of substances such as nutrients, but these substances have slow rates of diffusion. The enhancement of mixing by mechanical stirring mechanisms such as micro-swimmers has been investigated both experimentally [89, 93] and numerically [106, 163]. We are interested in mixing in microfluidic settings in which the flow complexity is driven by moving boundaries or suspensions of deformable particles. Mixing in flows with moving and deformable boundaries have been studied [22, 66, 129]. However, none of these works discuss mixing of vesicle suspensions.

Another important aspect in our work is the quantification of mixing. Although there is extensive work on metrics for mixing, there is not a universal measure [44, 92, 105, 113, 128]. We review some of the metrics specifically for advection. Metrics derived from dynamical systems consider the locations of tracer particles after a single period of a periodic flow. One example is the Poincaré section [10] which examines the position of particles after multiple periods of the flow. If the separation between neighboring particles increases exponentially with each period, then we say that the flow is chaotic, and the exponent, which is called the stretching rate or Lyapunov exponent, quantifies the mixing. In particular, larger Lyapunov exponents correspond to better mixing, and this approach is used in [122]. Mixing can also be measured statistically. One measure is the mixing variance metric. Another measure is the Kolmogorov-Sinai entropy that computes an integral of Lyapunov exponents over a domain [30]. Another set of metrics is based on tracking the interface between the solute and the solvent. When an effective mixing takes place, this interface grows rapidly. The exponential rate of the growth is called the interface stretch [116] and measures global stretching unlike

Lyapunov exponents [2, 6]. The aforementioned metrics are appropriate to measure mixing for advection dominated flows. However, the introduction of diffusion further enhances mixing. We refer the reader to [43] for a more detailed discussion on the different metrics for different Peclet numbers and initial conditions. When quantifying mixing due to diffusion, metrics that are based on the the solute are more informative. For example, the Euclidean (L^2) norm and the maximum norm (L^∞) [12, 152] can be used. However, L^p norms do not decay in the absence of diffusion, and therefore cannot quantify mixing due to advection. Thus, there is a need for a metric that captures mixing due to both diffusion and advection. One metric that captures mixing due to diffusion and advection is the negative index H^{-1} Sobolev norm [113, 114], which we will refer to as the *mix norm*. Additionally, Doering and Thiffeault [36], J.L.Thiffeault [75] compare the H^{-1} norm with L^p norms. In addition to being able to capture mixing due to advection and diffusion, the H^{-1} norm depends on the initial concentration field.

We use a Fourier-Chebyshev collocation [23, 162] method to discretize the advection-diffusion equation in space. We divide the transport equation into the diffusion and the advection equations using Strang operator splitting method [158]. We discretize the diffusion equation using the Crank-Nicolson scheme and the advection equation using the semi-Lagrangian method [150, 174].

Organization of the Chapter. We present the temporal and the spatial discretization methods for the advection-diffusion equation in Section 3.2. After we define the mixing metrics in Section 3.3, we show the results of the numerical experiments and discuss the

effects of area fraction, viscosity contrast, and the initial condition of the transported quantity on the mixing efficiency in Section 3.4.

Notation. We summarize the main notation used in this chapter in Table 3.1.

Table 3.1: *List of frequently used notation in Chapter 3.*

Symbol	Definition
Pe	Peclet number (3.1)
AF	Area fraction of vesicles
VC	Viscosity contrast
ϕ_0	Concentration in the absence of vesicles
ϕ	Concentration in the presence of vesicles
η	Mixing efficiency: ratio of $\|\|\phi_0\|\|$ to $\|\|\phi\|\|$
$\mathbf{v}_0$	Velocity field of a Couette flow without vesicles
$\mathbf{v}$	Velocity field of a vesicle suspension

3.2 Advection-diffusion Equation

The advection-diffusion equation governs mixing of a passive scalar. Its nondimensional form with a Neumann boundary condition is

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = \frac{1}{\mathrm{Pe}} \Delta \phi \quad \text{in } \Omega, \tag{3.3a}$$

$$\frac{\partial \phi}{\partial r} = 0 \quad \text{on } \Gamma. \tag{3.3b}$$

Here, ϕ is the concentration, $\mathbf{v}$ is the velocity, t is time, Ω is the domain between the inner and the outer cylinders in the Couette apparatus, and Γ is the boundary of Ω (the inner and the outer cylinders). In order to obtain $\mathbf{v}$ for flows with vesicles, we, first, simulate vesicle suspensions in a Taylor-Couette flow to obtain the positions and

tension of the vesicles and the hydrodynamic density on the fixed boundaries using the integral equation scheme described in Section 2.3. We, then, compute the velocity $\mathbf{v}$ at any point in the domain using the first equation in (2.10) as a postprocessing step.

3.2.1 Temporal Discretization

The Strang splitting method divides the solution operator $\mathscr{L}$ in (3.3a) into the advection ($\mathscr{L}_A$) and the diffusion ($\mathscr{L}_D$) operators:

$$\frac{\partial \phi}{\partial t} + \mathscr{L}_A \phi = 0, \tag{3.4a}$$

$$\frac{\partial \phi}{\partial t} + \mathscr{L}_D \phi = 0. \tag{3.4b}$$

Here, $\mathscr{L}_A = \mathbf{v} \cdot \nabla$ and $\mathscr{L}_D = -\frac{1}{\mathrm{Pe}}\Delta$. For the concentration field at time t^n, the Strang splitting evaluates the concentration at t^{n+1} in three steps: first it solves the advection equation (3.4a) in $[t^n, t^{(n+1)/2}]$, second the diffusion equation (3.4b) in $[t^n, t^{n+1}]$, and third the advection equation (3.4a) in $[t^{(n+1)/2}, t^{n+1}]$. This splitting is second-order accurate, but the methods used to solve each subproblem also determines the accuracy of the complete scheme. In this study, we solve the advection problem (3.4a) and the diffusion problem (3.4b) with the semi-Lagrangian method and the Crank-Nicolson method, respectively. This decoupling results in an unconditionally stable scheme [27].

Semi-Lagrangian for Advection. The advection equation (3.4a) in Lagrangian form is

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0,$$

which means that ϕ is constant along the characteristic path $\mathbf{x}(t)$ which satisfies

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}, t). \tag{3.5}$$

In the semi-Lagrangian method, first we solve (3.5) backward in time to find the Lagrangian point or *departure point* $\mathbf{x}_d$ that arrives at a point $\mathbf{x}_a$. The arrival point $\mathbf{x}_a$ coincides with the discretization points used for the diffusion solve (see Fig. 3.3). This trajectory is computed with the second-order explicit midpoint rule

$$\mathbf{x}_m = \mathbf{x}_a - \mathbf{v}(\mathbf{x}_a, t^n)\frac{\Delta t_A}{2}, \tag{3.6a}$$

$$\mathbf{x}_d = \mathbf{x}_a - \mathbf{v}\left(\mathbf{x}_m, t^n + \frac{\Delta t_A}{2}\right)\Delta t_A. \tag{3.6b}$$

Here Δt_A is the time step size for the advection problem (usually $\Delta t_D \geq \Delta t_A$ where Δt_D is the time step size for the diffusion problem). Since we integrate along the characteristic line, the concentration at $\mathbf{x}_a$ satisfies $\phi\left(\mathbf{x}_a, t^{n+1}\right) = \phi\left(\mathbf{x}_d, t^n\right)$. In general, the departure points $\mathbf{x}_d$ do not coincide with the grid points, thus we interpolate the concentration at $\mathbf{x}_d$ using cubic interpolation with $\phi(\mathbf{x}_a, t)$. Additionally, we also interpolate the velocity at the mid-point $\mathbf{x}_m$ in (3.6b) using the same method. This particular semi-Lagrangian scheme is second-order accurate in time [39, 174].

Crank-Nicolson for Diffusion. We use the Crank-Nicolson scheme to discretize the diffusion equation (3.4b) in time

$$\frac{\phi^{n+1} - \phi^n}{\Delta t_D} = \frac{1}{\mathrm{Pe}}\Delta\left(\frac{\phi^{n+1} + \phi^n}{2}\right). \tag{3.7}$$

Since (3.7) is not L-stable, high frequency components of ϕ can lead to spurious numerical oscillations [29]. Since we choose discrete initial conditions (see Fig. 3.1(b)), high

frequency components will be present. Therefore, we require a method that behaves as a numerical low-pass filter so that high frequencies are suppressed [103]. We apply the L-stable backward Euler method initially to smooth the initial condition [170, 185]. Since backward Euler is first-order accurate, it is only applied for $t \in [0, \Delta t_D]$ with a time step size $\Delta t^{BE} = \Delta t_D^2$. Then, to achieve second-order accuracy, Crank-Nicholson is used for $t > \Delta t_D$.

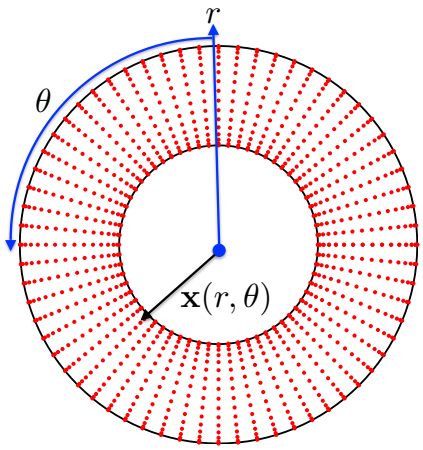


Figure 3.3: *Fourier-Chebyshev grid for the advection-diffusion equation.*

3.2.2 Spatial Discretization

Taken advantage of symmetries in the geometry, we use polar coordinates (r, θ) and a pseudo-spectral representation of ϕ. Since ϕ is periodic in θ, we use a Fourier series in θ

$$\phi(r, \theta, t) = \sum_{k=-N_\theta/2+1}^{N_\theta/2} \widehat{\phi}_k(r, t) e^{ik\theta}. \tag{3.8}$$

Then, we discretize the Fourier coefficients $\widehat{\phi}_k$ in r using Chebyshev polynomials as

$$\widehat{\phi}_k(r,t) = \sum_{m=0}^{N_r-1} \widehat{\phi}_{k,m}(t)\cos m\alpha.$$

Here, N_θ is the number of uniformly distributed collocation points in $\theta \in [0, 2\pi]$ and N_r is the number of collocation points in r. Additionally, $\alpha = \pi m/(N_r - 1) \in [0, \pi]$ and we define the radial coordinate as $r = \frac{1}{2}(1-\cos\alpha)(r_2 - r_1) + r_1$, where the radii of the inner and the outer cylinders are r_1 and r_2, respectively. The resulting grid points are illustrated in Fig. 3.3. After substituting (3.8) into (3.7) and applying the Laplacian Δ, the resulting diagonal set of linear equations is

$$\mathscr{A}_k^- \widehat{\phi}_k^{n+1}(r) = \mathscr{A}_k^+ \widehat{\phi}_k^{n}(r), \tag{3.9}$$

where the operators $\mathscr{A}_k$ are

$$\mathscr{A}_k^\mp = \frac{I}{\Delta t} \mp \frac{1}{2\mathrm{Pe}}\left(\frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial r^2} + \frac{k^2}{r^2}\right),$$

and I is the identity matrix. Equation (3.9) is efficiently solved using the fast cosine transform.

We have tested our numerical scheme on different initial conditions and velocity fields. For smooth velocity fields and initial conditions, the method is second-order accurate in time and spectrally accurate in space. We have also tested our solver on velocity fields that are not continuous, such as those for vesicle suspensions, and we achieve similar convergence rates for smooth initial conditions.

3.3 Mixing Metrics

To measure mixing in advection-dominated transport, some of the early work [75, 113] suggests that the H^{-1} norm is appropriate and discusses the disadvantages of L^p norms. In this section, we define and compare the L^1, L^2, and H^{-1} norms on an example problem.

The L^p norm of the concentration ϕ is

$$\|\phi\|_{L^p} = \left(\int_\Omega |\phi(\mathbf{x})|^p d\Omega \right)^{1/p}.$$

We only use $p = 1, 2$. The H^{-1} norm of a concentration field ϕ is a negative Sobolev norm and is defined as

$$\|\phi\|_{H^{-1}} = \left(\int_\Omega g(\mathbf{x})\phi(\mathbf{x}) d\mathbf{x} \right)^{1/2},$$

where g is the solution of the boundary value problem

$$\begin{aligned} (I - \Delta)\, g(\mathbf{x}) &= \phi(\mathbf{x}) \qquad \mathbf{x} \in \Omega, \\ g(\mathbf{x}) &= 0 \qquad\qquad \mathbf{x} \in \Gamma, \end{aligned} \tag{3.10}$$

where I is the identity operator and Δ is the Laplacian. We solve (3.10) using the Fourier-Chebyshev collocation scheme described in Section 3.2.2. In L^2 and H^{-1}, smaller norms of ϕ correspond to a more mixed concentration field. In advection-dominated flows, in which the diffusivity D is small, the concentration gradient can become large enough that the L^2 norm will decay (see Fig. 3.4(d)) [75]. However, if D is sufficiently small, this will not provide a meaningful quantification for mixing. In contrast, the H^{-1} norm captures mixing due to both advection and diffusion which

results in a more accurate method for measuring mixing for a large range of Peclet numbers (see Figures 3.4(c) and 3.4(d)). Since we are interested in advection-dominated transport, the H^{-1} norm is preferable to the L^2 norm to quantify mixing and to define the mixing efficiency (3.2). If the efficiency $\eta > 1$, then $\|\phi_0\| > \|\phi\|$ and thus the presence of vesicles promotes mixing. Similarly, if $\eta < 1$, then the presence of vesicles suppresses mixing.

Remark. By integrating (3.3a) over Ω, integrating by parts, and applying the Neumann boundary condition (3.3b) and the incompressibility constraint, we have

$$\frac{d}{dt}\int_\Omega |\phi(\mathbf{x},t)|d\mathbf{x} = 0.$$

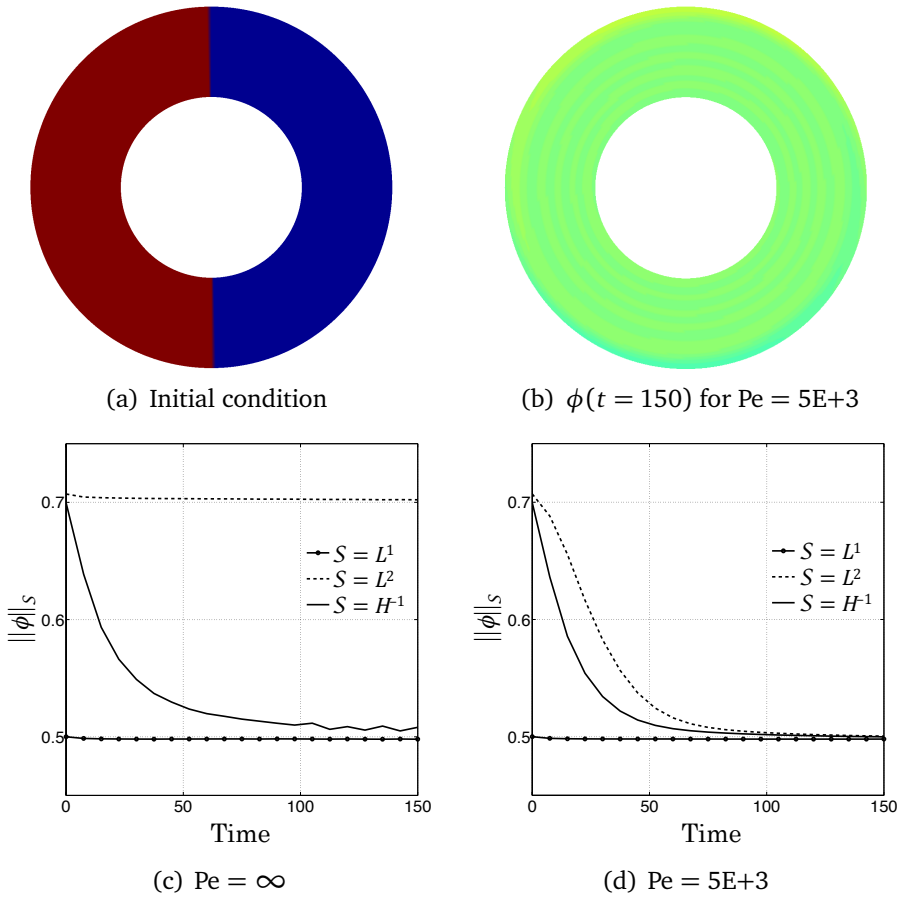
Here we have used the fact that the concentration field is positive. Since the completely mixed state corresponds to a uniform concentration $\overline{\phi}$, we have

$$\int_\Omega |\phi(\mathbf{x},t)|d\mathbf{x} = \overline{\phi}\int_\Omega d\mathbf{x},$$

for all time. Therefore the L^1 norm is not an appropriate norm to measure mixing. The L^2 norm measures fluctuations from the mean $\overline{\phi}$. It has been shown that the L^2 norm of the concentration monotonically approaches zero in the absence of sources since [75]

$$\frac{d}{dt}\|\phi\|_{L^2}^2 = -2D\|\nabla\phi\|_{L^2}^2.$$

Therefore, if the diffusion coefficient is non-zero, or $Pe \neq \infty$, and the concentration field is not uniform, then the L^2 norm of the concentration field decreases with time. We demonstrate these properties of the L^1 and L^2 norm in Fig. 3.4. The initial condition is



(a) Initial condition (b) $\phi(t = 150)$ for Pe = 5E+3

(c) Pe $= \infty$ (d) Pe = 5E+3

Figure 3.4: *Comparison of mixing metrics. We illustrate the differences between L^1, L^2, and H^{-1} norms for the mixing of the initial condition (a). The underlying flow is a Couette flow without vesicles. We consider two Peclet numbers* Pe $= \infty$ *and* Pe *=5E+3. We do not normalize the norms so as to demonstrate that the norms converge to the same value as $t \to \infty$. (c) for* Pe $= \infty$ *the L^1 and L^2 norms do not decay without diffusion, even though mixing is taking place by advection. (d) for* Pe *=5E+3 as the concentration becomes uniform over the domain as in (b), the L^2 and the H^{-1} norms approach one another and ultimately converge to the constant L^1 norm.*

depicted in Fig. 3.4(a) and we consider a Couette flow without vesicles. The simulation parameters are in Table 3.2. First, Fig. 3.4(c) shows that neither the L^1 norm nor the L^2 are able to capture mixing due to pure advection (i.e., Pe $= \infty$), while the H^{-1}

norm decays as the concentration is mixed. Second, Fig. 3.4(d) demonstrates that in the presence of diffusion, the L^1 norm is still independent of time, but the L^2 and H^{-1} norms decrease with mixing. Additionally, in the presence of diffusion, as the concentration becomes uniform, the three norms approach the same value.

3.4 Numerical Experiments

In this section, we discuss the effects of the presence of vesicles on mixing. We investigate the dependence of the mixing efficiency on the area fraction (AF) and the viscosity contrast (VC). Additionally, we run tests on the different initial conditions shown in Fig. 3.1(b). We summarize the experiments as follows.

- **Effects of area fraction (Fig. 3.6, Fig. 3.7 and Fig. 3.8):** We aim to understand whether the presence of more vesicles in a suspension promotes or suppresses mixing. For this purpose, we simulate mixing in vesicle suspensions with area fractions 10%, 20%, and 40%, and with the layer initial condition and Pe = 1E+4. We visualize several time snapshots of the concentration field, the vesicle positions, and the velocity field (in fact, we visualize its *difference* from the pure Couette flow). The results indicate, for this setup, that increasing the area fraction promotes mixing.

- **Effects of Peclet number and initial condition (Fig. 3.9 and Fig. 3.10):** We simulate mixing in the vesicle suspensions with three area fractions of 10%, 20%, and 40% with various Peclet numbers and initial conditions (Fig. 3.1(b)). In order to quantify the effect of the Peclet number and initial condition, we plot

the mixing efficiency η with respect to time. The results show that the presence of vesicles manifests its effects on mixing at very high Peclet numbers. However, depending on the initial condition, the vesicles promote (layer initial condition), suppress (dye initial condition), and do not affect (random initial condition) mixing.

- **Effects of viscosity contrast (Fig. 3.13):** We consider vesicle suspensions with area fractions 5% and 10% and various viscosity contrasts. We perform the viscosity contrast tests on the layer and dye initial conditions. At higher viscosity contrasts, the vesicles undergo the more complex dynamics (tumbling) when compared to that (tank-treading) at lower viscosity contrasts. The result is more complicated stirring which increases the mixing efficiency. However, the viscosity contrast has less of an effect on the efficiency when compared to the area fraction.

We consider vesicles of reduced area 0.65. The inner boundary rotates at a constant angular velocity while the outer boundary is fixed. The inner boundary of the simulations with area fractions of 5% and 40% completes 32 rotations, while it completes 21 rotations in all other simulations. In all the runs, the vesicles are discretized with 64 points, and in all but one run, the outer cylinder is discretized with 128 points. For the simulation with 40% area fraction, the outer cylinder is discretized with 256 points.

We list the parameters and their values used in the experiments in Table 3.2. We discretize the transport equation with $N_r = 256$ collocation points in the radial

direction and with $N_\theta = 1024$ collocation points in the azimuthal direction. The time step sizes for the diffusion (3.7) and the advection (3.6) equations are $\Delta t_D = 0.04$ and $\Delta t_A = 0.01$, respectively.

Table 3.2: *Parameters for mixing simulations.*

	Notation	Value	Unit
Angular velocity of the inner cylinder	ω	0.67	rad/ms
Inner cylinder's radius	r_1	30	μm
Outer cylinder's radius	r_2	60	μm

3.4.1 Statistical Analysis

Physically meaningful experiments requires vesicle suspensions to become statistically stationary. Here we explain how we detect the statistical stationarity in this study. As we mentioned earlier, the velocity from the vesicle simulations are used to drive the advective part of mixing. We use statistics of the velocity field to determine the onset of the statistical equilibrium. We start the mixing simulation once these statistics, which we will define shortly, become time independent.

The presence of vesicles perturbs the velocity field of the default Couette flow. We define the perturbation $\tilde{\mathbf{v}}$ as $\mathbf{v} = \mathbf{v}_0 + \tilde{\mathbf{v}}$, where $\mathbf{v}$ is the velocity field of the vesicle suspension and $\mathbf{v}_0$ is the velocity field in the absence of vesicles. We monitor stationarity of the time series of $\nu(t) = \|\tilde{\mathbf{v}}\|_{L^2}$. A stationary time series has statistical properties that do not change over time, i.e.,its mean and variance over any statistically representative window remain the same. Our statistical analysis involves, first, determining the statistically representative window size w' of the time series and, second, finding the time when the statistical equilibrium is first reached. Given a time series $\nu(t)$, the

window size w' is chosen so that v between t to $t+w'$ has the same mean and variance at any time t. We describe the steps to find w' as follows.

- We obtain a number of samples of a window width w from the time series $v(t)$ starting at randomly chosen times $t_i \in [0, T_h - w]$ where T_h is the time horizon. We denote these samples with $v_{w,i} = (v(t_i), v(t_i + w))$.

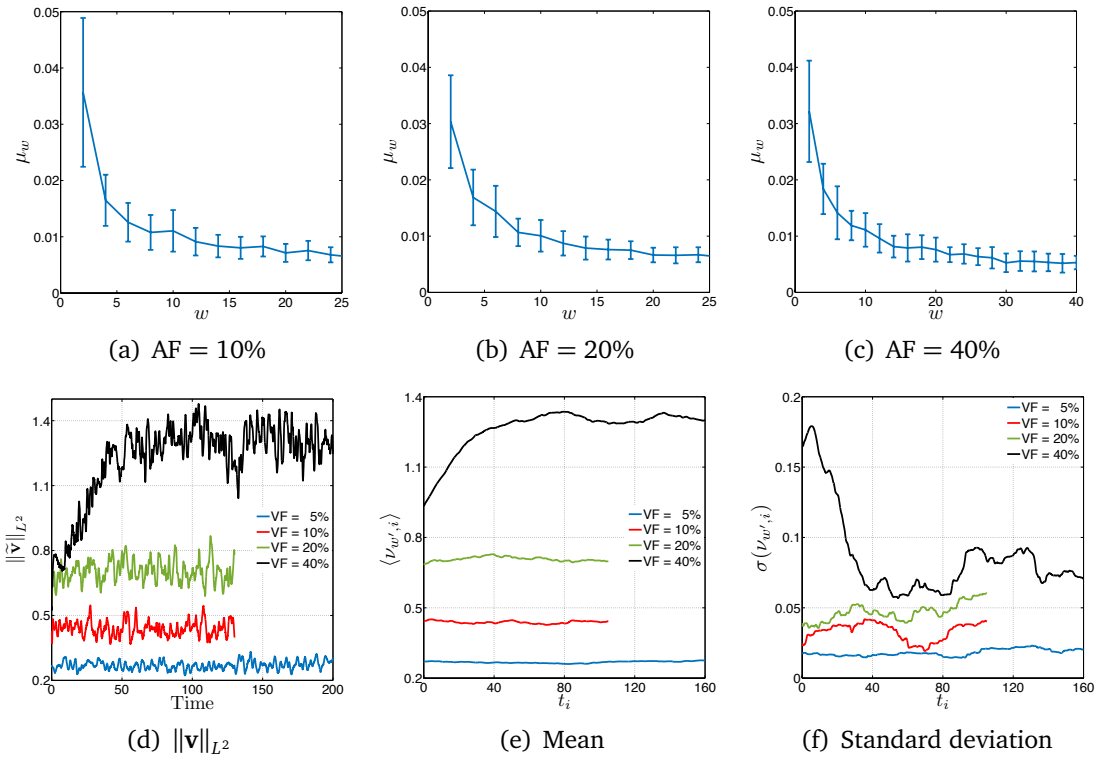
- We compute the Fourier transform of the oscillations of the mean $\tilde{v}_{w,i} = v_{w,i} - \langle v_{w,i} \rangle$, where $\langle \cdot \rangle$ denotes the mean value $\langle \cdot \rangle = \frac{1}{w} \int_{t_i}^{t_i+w} \cdot \, dt$. We then sum the Fourier coefficients to form $\mu_{w,i}$ which represents a property (in this case, the energy) of the particular window.

- As the window size w increases, the ensemble average $\mu_w = \frac{1}{M} \Sigma_{i=1}^{M} \mu_{w,i}$ converges to the mean of the entire time series, and the standard deviation, $\{\mu_{w,i}\}_{i=1}^{M}$, decreases, where M is the number of samples. The statistically representative window size w' is the one which delivers a small standard deviation and a mean close to that of the entire time series.

We plot the means of the samples as a function of window size w for the suspensions of different area fractions in Figures 3.5(a)- 3.5(c). We choose the window size $w' = 25$ time units for the suspensions at AF $= 10\%$ and 20%, and $w' = 40$ time units for those at AF $= 5\%$ and 40%. Although we do not show the results for the vesicle suspensions with VC $= 5$ and VC $= 8$, we perform the same analysis for them.

Once the representative window size w' of the time series $v(t)$ is chosen, we need to determine when the statistical equilibrium is first reached. To do so, we choose

samples $\nu_{w',i}$ of size w' at every discrete time $t_i \in [0, T_h - w']$ and compute their means $\langle \nu_{w',i} \rangle$ and standard deviations $\sigma(\nu_{w',i})$. We determine the time when the statistical equilibrium is attained by examining when the mean and standard deviation plateau. In particular, we require that $|\langle \nu_{w',i+1} \rangle - \langle \nu_{w',i} \rangle| / |\langle \nu_{w',i} \rangle|$ to be less than some tolerance. We present the statistical analysis results for the different area fractions at the bottom of Fig. 3.5. The mean (Fig. 3.5(e)) and the standard deviation (Fig. 3.5(f)) converge after $t_i = 100$ in AF $= 5\%$, $t_i = 40$ in AF $= 10\%$ and AF $= 20\%$, and $t_i = 100$ in AF $= 40\%$.



(a) AF = 10% (b) AF = 20% (c) AF = 40%

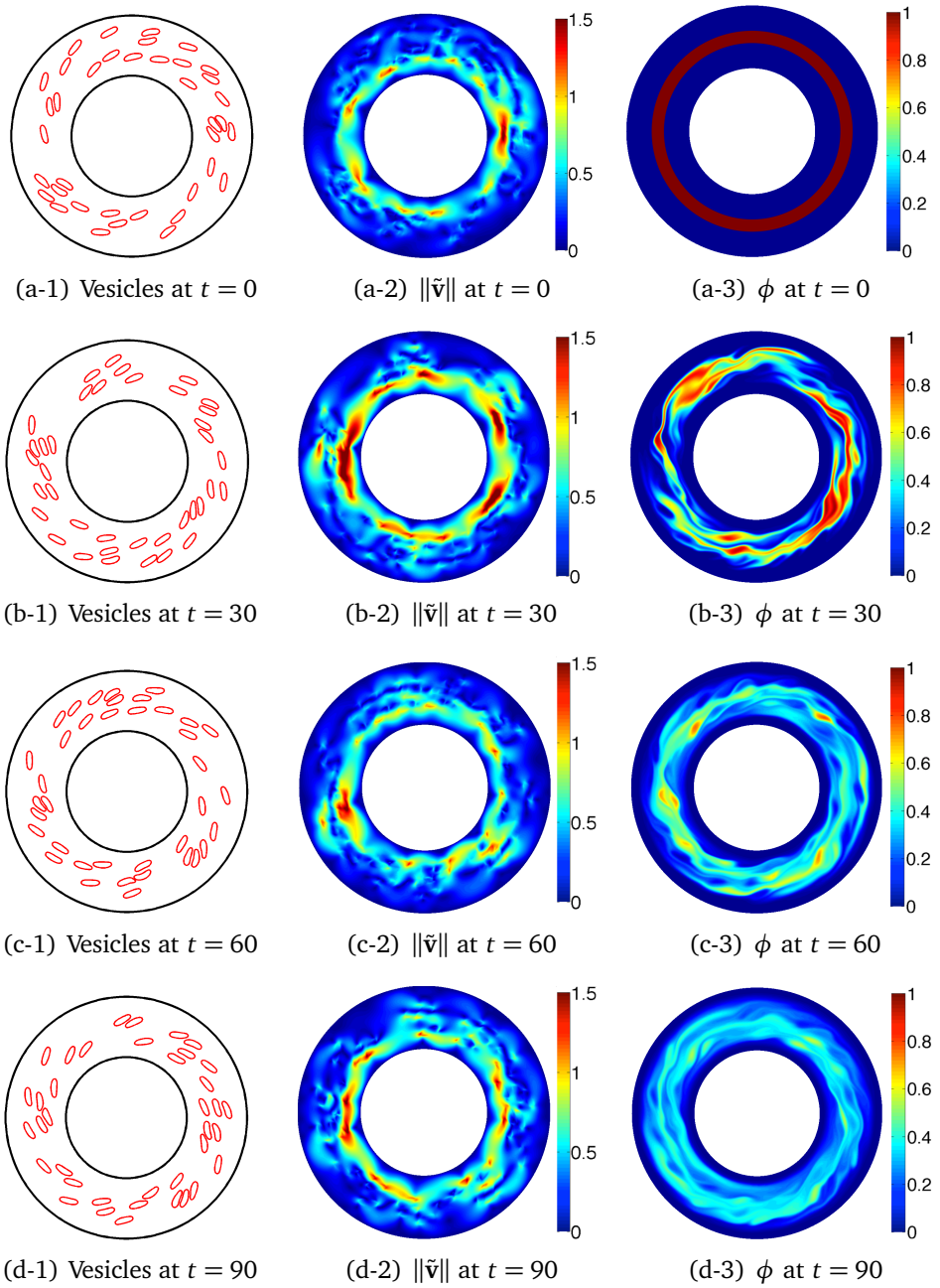
(d) $\|\mathbf{v}\|_{L^2}$ (e) Mean (f) Standard deviation

Figure 3.5: *We consider the L^2 norm of the perturbations $\tilde{\mathbf{v}}$ in (d) to examine the statistical stationarity of the vesicle suspensions. We compute the mean (e) and the standard deviation (f) of $\|\tilde{\mathbf{v}}\|_{L^2}$ in the window $(t_i, t_i + w')$, for all i. To find statistically representative window size, we look at the energy of randomly chosen samples from $\|\tilde{\mathbf{v}}\|_{L^2}$ of various window widths w for each area fraction in the top row.*

3.4.2 Effects of Area Fraction

We simulate mixing in vesicle suspensions with the area fractions 10%, 20%, and 40%, and with no viscosity contrast. We use the layer initial condition. We fix the Peclet number to Pe=1E+4 for all the area fractions by adjusting the diffusivity based on the value of $\langle \|\mathbf{v}\|_{L^2} \rangle$. We show the vesicle positions, the magnitude of the perturbation in the Couette flow due to the vesicles $\|\tilde{\mathbf{v}}\|$ and the concentration ϕ for the area fractions

10% in Fig. 3.6, 20% in Fig. 3.7, and 40% in Fig. 3.8. The results show that as the area fraction increases, the maximum value of $\|\tilde{\mathbf{v}}\|$ increases from approximately 1.5 to 4 wherein the maximum of the magnitude of the nondimensional velocity is 10 (see the second columns in Fig. 3.6 and Fig. 3.8). The corresponding concentration fields observably differ as the area fraction of the vesicles increases (see the third columns in Fig. 3.6 and Fig. 3.8). In addition to the qualitative results in Fig. 3.6, Fig. 3.7, and Fig. 3.8, the first row in Fig. 3.9 demonstrates the corresponding mixing efficiencies. We see that the presence of vesicles enhances mixing for this particular initial condition and increasing the area fraction to 40% increases the efficiency up to $\eta \approx 1.35$.



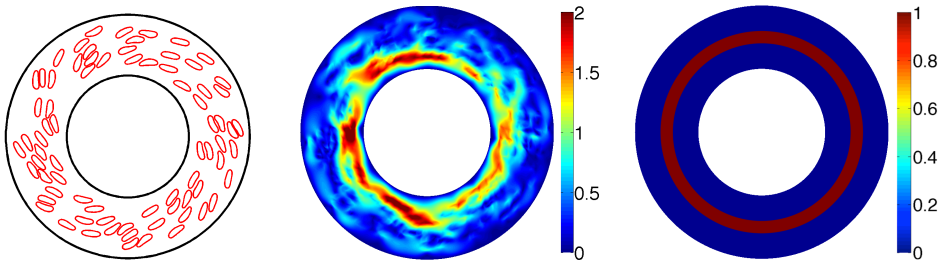
(a-1) Vesicles at $t = 0$ (a-2) $\|\tilde{\mathbf{v}}\|$ at $t = 0$ (a-3) ϕ at $t = 0$

(b-1) Vesicles at $t = 30$ (b-2) $\|\tilde{\mathbf{v}}\|$ at $t = 30$ (b-3) ϕ at $t = 30$

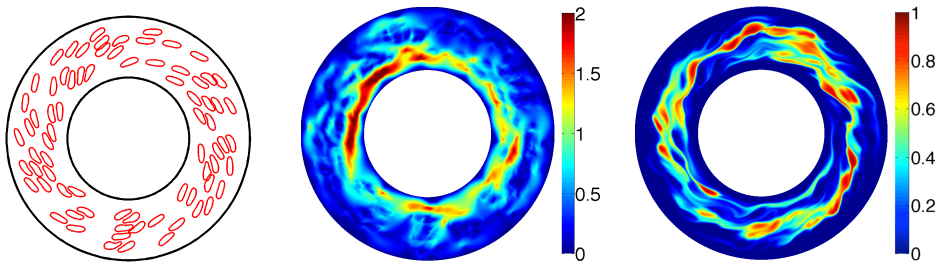
(c-1) Vesicles at $t = 60$ (c-2) $\|\tilde{\mathbf{v}}\|$ at $t = 60$ (c-3) ϕ at $t = 60$

(d-1) Vesicles at $t = 90$ (d-2) $\|\tilde{\mathbf{v}}\|$ at $t = 90$ (d-3) ϕ at $t = 90$

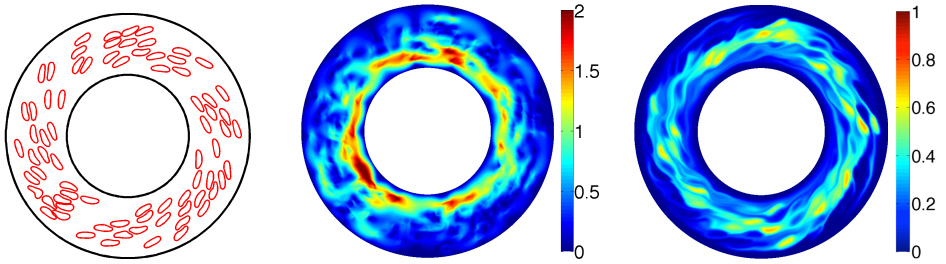
Figure 3.6: *Effects of 10% area fraction. We present the vesicle positions (left), the magnitude of the velocity field due only to the vesicles* $\|\tilde{\mathbf{v}}\|$ *(middle), and the concentration* ϕ *(right) for the layer initial condition.*



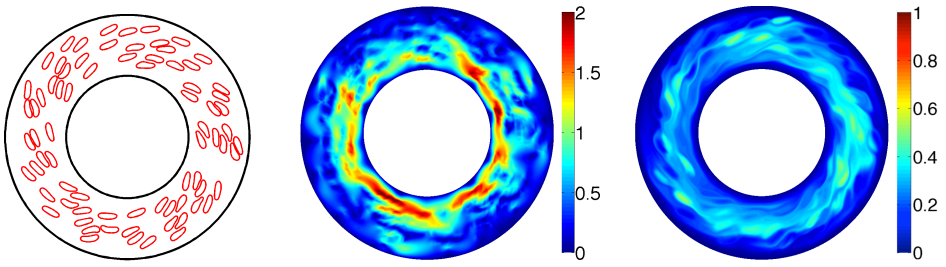
(a-1) Vesicles at $t = 0$ (a-2) $\|\tilde{\mathbf{v}}\|$ at $t = 0$ (a-3) ϕ at $t = 0$



(b-1) Vesicles at $t = 30$ (b-2) $\|\tilde{\mathbf{v}}\|$ at $t = 30$ (b-3) ϕ at $t = 30$

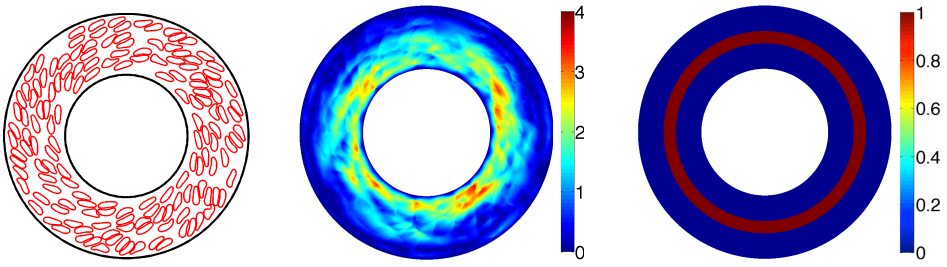


(c-1) Vesicles at $t = 60$ (c-2) $\|\tilde{\mathbf{v}}\|$ at $t = 60$ (c-3) ϕ at $t = 60$

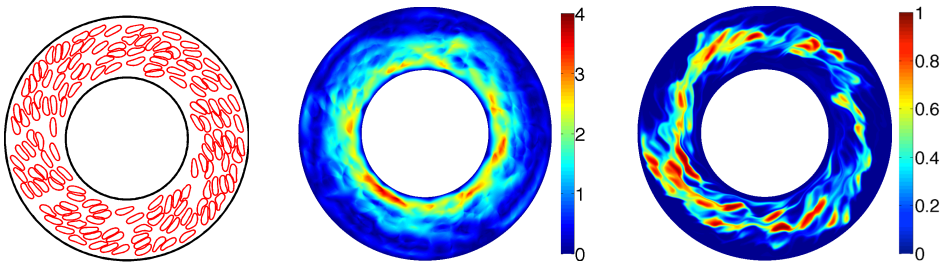


(d-1) Vesicles at $t = 90$ (d-2) $\|\tilde{\mathbf{v}}\|$ at $t = 90$ (d-3) ϕ at $t = 90$

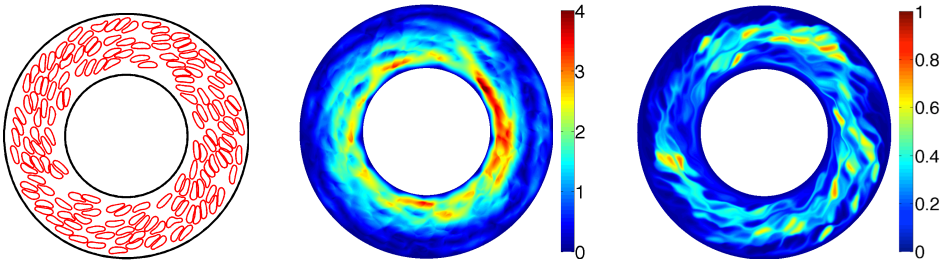
Figure 3.7: *Effects of 20% area fraction. We present the vesicle positions (left), the magnitude of the velocity field due only to the vesicles* $\|\tilde{\mathbf{v}}\|$ *(middle), and the concentration* ϕ *(right) for the layer initial condition.*



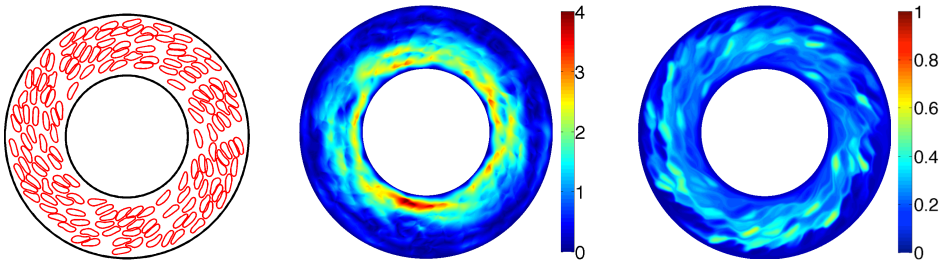
(a-1) Vesicles at $t = 0$ (a-2) $\|\tilde{\mathbf{v}}\|$ at $t = 0$ (a-3) ϕ at $t = 0$



(b-1) Vesicles at $t = 30$ (b-2) $\|\tilde{\mathbf{v}}\|$ at $t = 30$ (b-3) ϕ at $t = 30$



(c-1) Vesicles at $t = 60$ (c-2) $\|\tilde{\mathbf{v}}\|$ at $t = 60$ (c-3) ϕ at $t = 60$



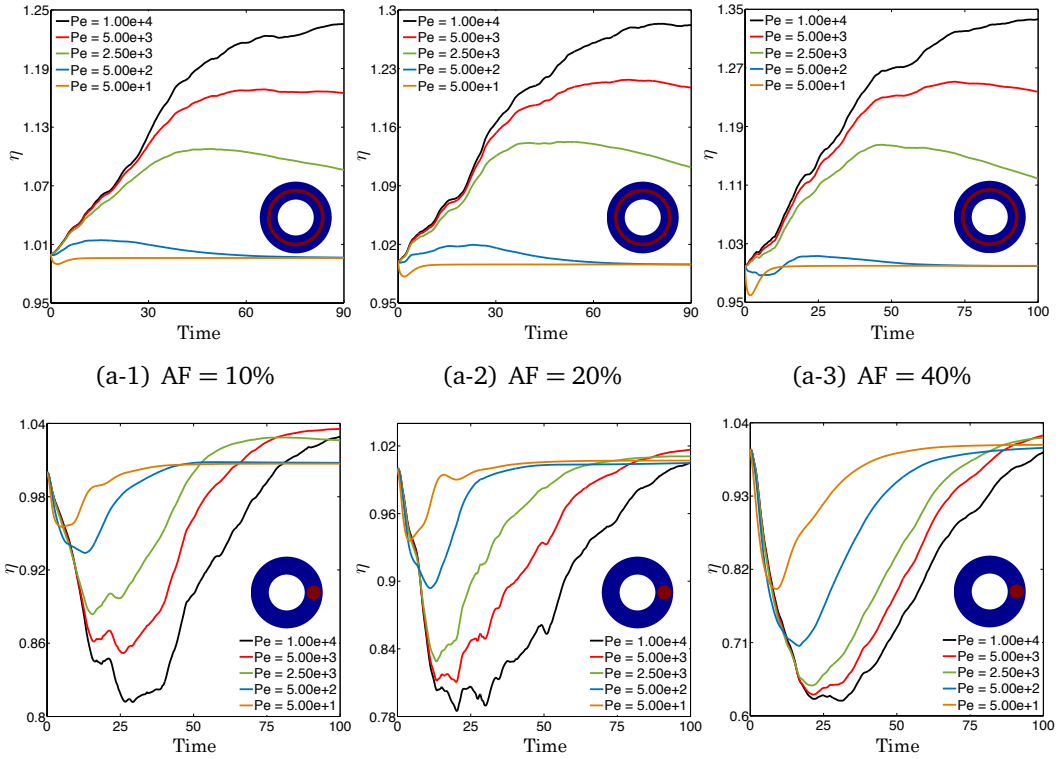
(d-1) Vesicles at $t = 90$ (d-2) $\|\tilde{\mathbf{v}}\|$ at $t = 90$ (d-3) ϕ at $t = 90$

Figure 3.8: *Effects of 40% area fraction. We present the vesicle positions (left), the magnitude of the velocity field due only to the vesicles* $\|\widetilde{\mathbf{v}}\|$ *(middle), and the concentration* ϕ *(right) for the layer initial condition.*

3.4.3 Effects of Peclet Number and Initial Condition

We simulate mixing with the Peclet numbers 1E+4, 5E+3, 2.5E+3, 5E+2 and 5E+1, and for all four initial conditions in Fig. 3.1(b). We, then, demonstrate the mixing efficiency η with respect to time in Fig. 3.9 and Fig. 3.10. The results in Fig. 3.9 and Fig. 3.10 show that the mixing efficiency is close to one for $\mathrm{Pe} = \mathcal{O}(10)$, but it deviates from one as the Peclet number increases. This is expected since the perturbations in the velocity field become more important as the transport becomes more advection-dominated. While, the first row in Fig. 3.9 shows that the presence of vesicles enhances mixing (i.e. $\eta > 1$) for the layer initial condition, the second row demonstrates that vesicles suppress mixing for the dye initial condition. Additionally, as the area fraction increases (from left to right in Fig. 3.9 and Fig. 3.10), the maximum efficiency increases for the layer initial condition and the minimum efficiency decreases for the dye initial condition. The first row in Fig. 3.10 illustrates that mixing is better in the absence of vesicles for the vesicle initial condition, however, the effects of the vesicles on mixing becomes less important as the area fraction increases. Furthermore, a Couette flow without vesicles provides the same quality of mixing as the one with vesicles for the random initial condition for any area fraction (see the second row in Fig. 3.10).

We observe that in the presence of vesicles, mixing is promoted for the layer initial condition, while it is suppressed for the other three initial conditions. In Fig. 3.11 and Fig. 3.12, we plot the concentration field ϕ with $\mathrm{Pe} = 10^4$ at a series of time steps when the dye, vesicle and random initial conditions are used. The flow in Fig. 3.11 includes vesicles with no viscosity contrast at 40% area fraction while the flow in Fig. 3.12 is the default Couette flow without vesicles. In Fig. 3.11 and Fig. 3.8, we see

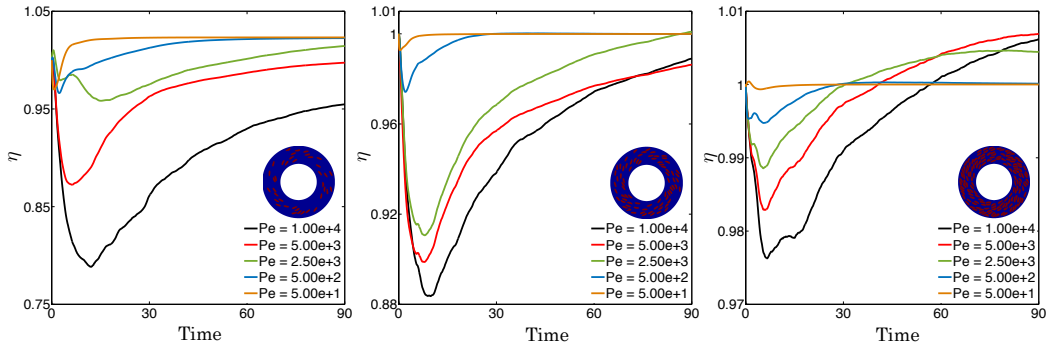


(a-1) AF = 10% (a-2) AF = 20% (a-3) AF = 40%

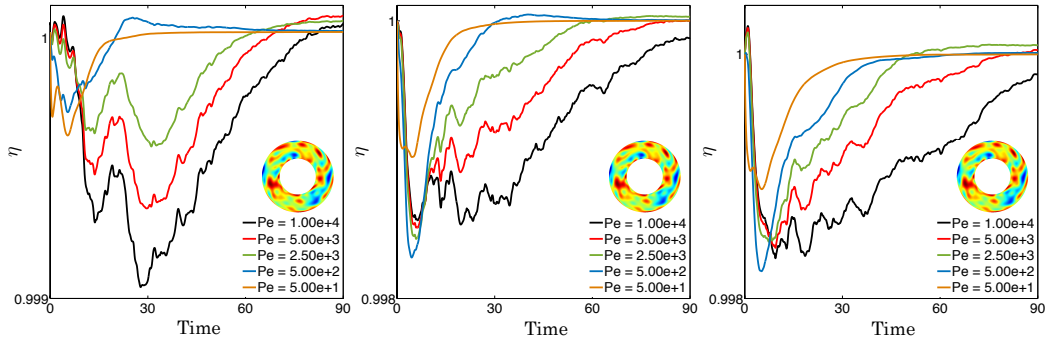
(b-1) AF = 10% (b-2) AF = 20% (b-3) AF = 40%

Figure 3.9: *Effects of the area fraction on mixing for various Peclet numbers and initial conditions. The first row demonstrates that the mixing efficiency η increases with increasing area fraction of the vesicles for the layer initial condition. The second row shows that η decreases when the initial condition is switched to the dye.*

that the vesicle dynamics lead to a complex stirring pattern, but the vesicles create trapped regions of the concentration. However, in Fig. 3.12, we also see a complex stirring pattern for the dye, vesicle, and random initial concentrations in the absence of vesicles, but without the trapped regions. Therefore, as observed in the bottom row of Fig. 3.9 and Fig. 3.10, the mixing efficiency is less than 1. However, for the layer initial condition, in the absence of vesicles, the transport term $\mathbf{v} \cdot \nabla \phi$ vanishes since



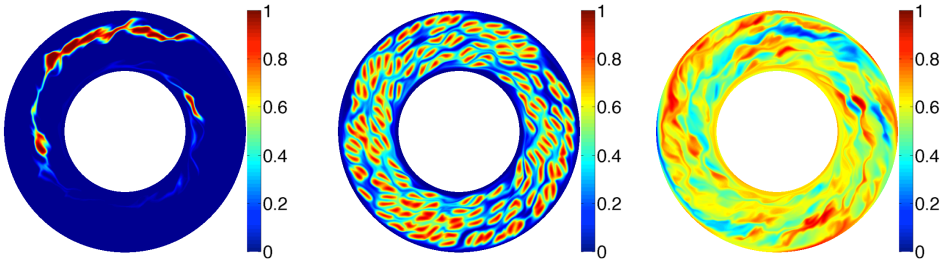
(c-1) AF = 10% (c-2) AF = 20% (c-3) AF = 40%



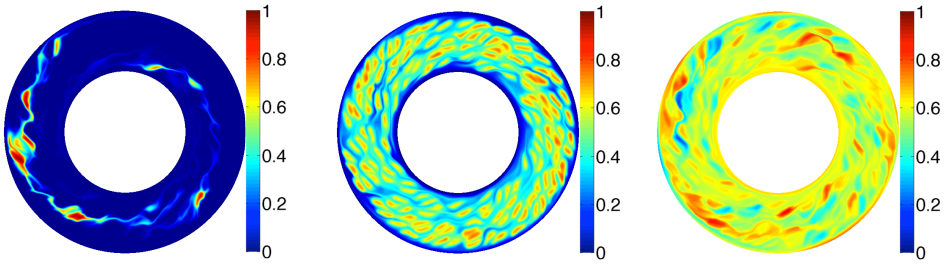
(d-1) AF = 10% (d-2) AF = 20% (d-3) AF = 40%

Figure 3.10: *Effects of the area fraction on mixing for various Peclet numbers and the initial conditions. The first row shows the mixing efficiency η with respect to time for the vesicle initial conditions for each area fractions and the second row is for the random initial condition. The results show that there is no clear effect of the presence mixing for the random initial condition.*

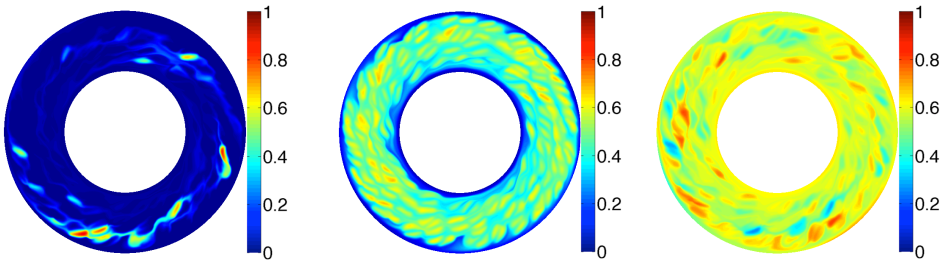
the gradient of the initial condition only has a radial component, while the default Couette flow does not have a radial component. Therefore, mixing can occur only due to diffusion. Since the Peclet number is large, little mixing occurs due to diffusion without the vesicles, and the mixing efficiency turns out to be greater than 1 as observed in the top row of Fig. 3.9. In Section 3.4.5, we define a mixing measure M that can be used to predict the mixing efficiency of a given initial condition.



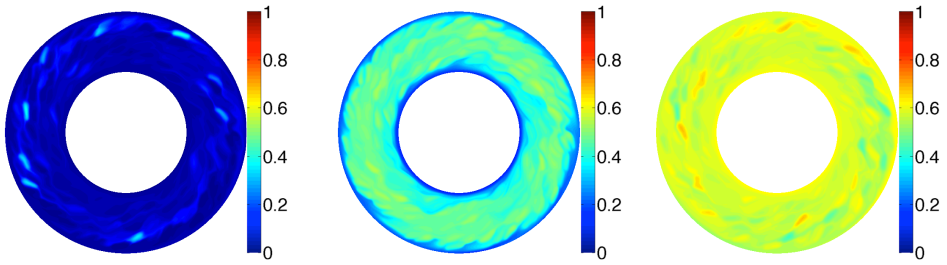
(a-1) *DYE* at $t = 15$ (b-1) *VESICLE* at $t = 15$ (c-1) *RANDOM* at $t = 15$



(a-2) *DYE* at $t = 30$ (b-2) *VESICLE* at $t = 30$ (c-2) *RANDOM* at $t = 30$

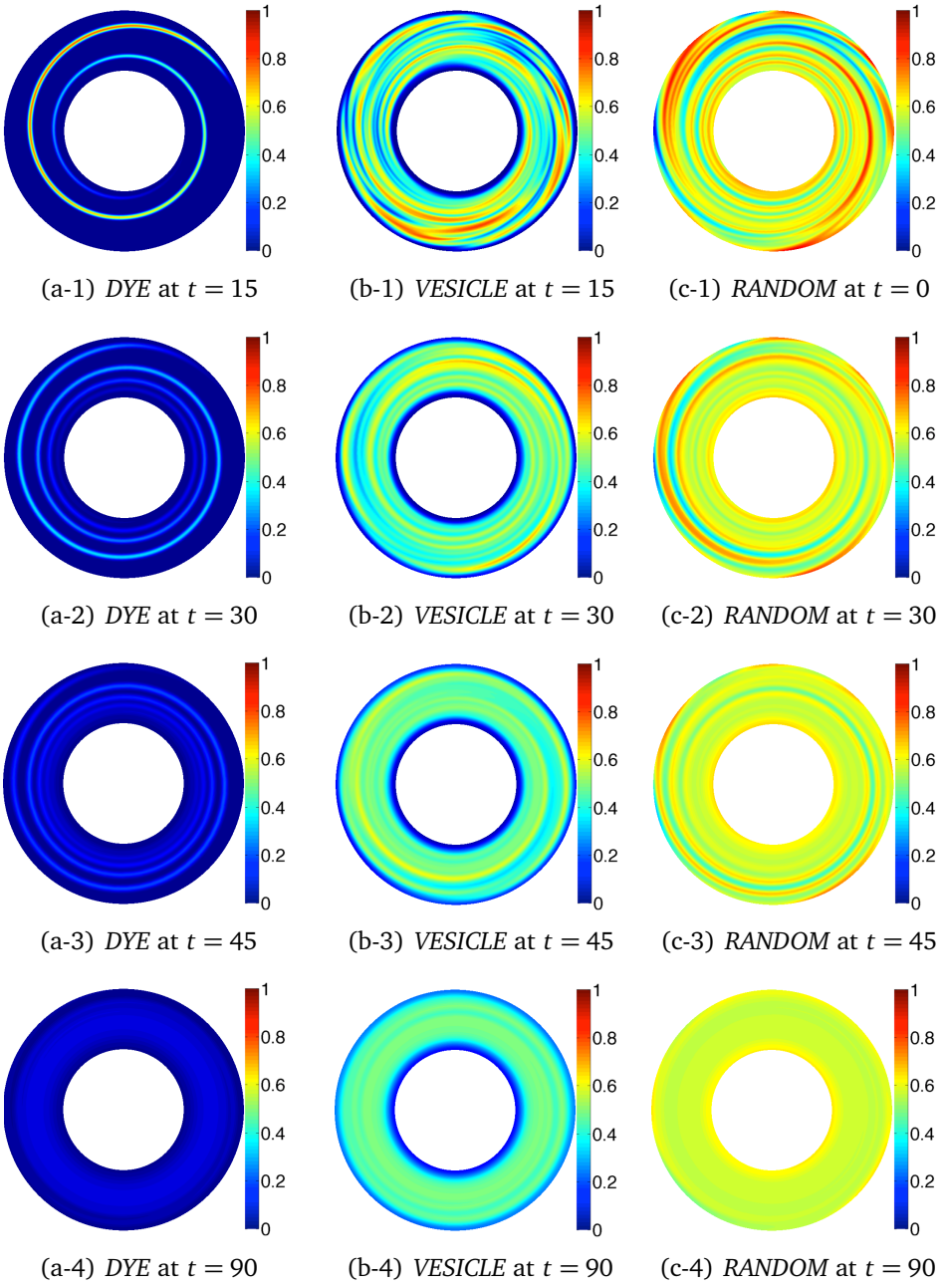


(a-3) *DYE* at $t = 45$ (b-3) *VESICLE* at $t = 45$ (c-3) *RANDOM* at $t = 45$



(a-4) *DYE* at $t = 90$ (b-4) *VESICLE* at $t = 90$ (c-4) *RANDOM* at $t = 90$

Figure 3.11: *Evolution of the concentration field ϕ for various initial conditions in the presence of vesicles. We present the evolution of DYE (left), VESICLE (middle) with 40% area fraction, and RANDOM (right) initial concentrations at $Pe = 10^4$. The velocity field is post-processed from the vesicle simulation with $AF = 40\%$ and $VC = 1$. See Fig. 3.12 for mixing without vesicles.*



(a-1) *DYE* at $t = 15$ (b-1) *VESICLE* at $t = 15$ (c-1) *RANDOM* at $t = 0$

(a-2) *DYE* at $t = 30$ (b-2) *VESICLE* at $t = 30$ (c-2) *RANDOM* at $t = 30$

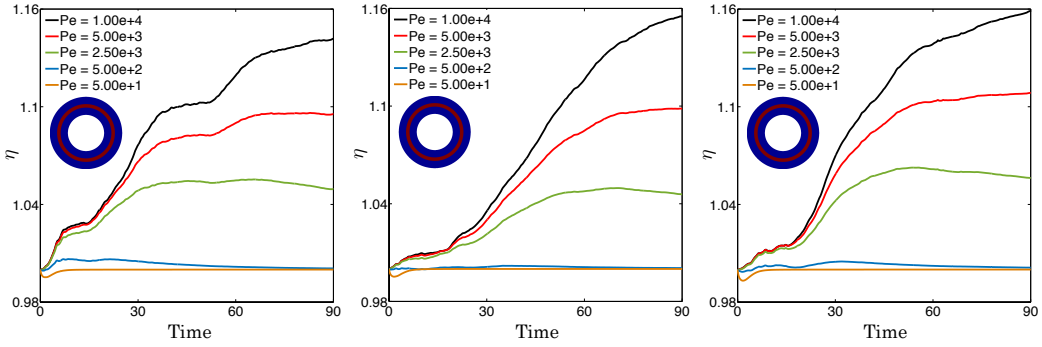
(a-3) *DYE* at $t = 45$ (b-3) *VESICLE* at $t = 45$ (c-3) *RANDOM* at $t = 45$

(a-4) *DYE* at $t = 90$ (b-4) *VESICLE* at $t = 90$ (c-4) *RANDOM* at $t = 90$

Figure 3.12: *Evolution of the concentration field ϕ for various initial conditions in the absence of vesicles. We present the evolution of DYE (left), VESICLE (middle) with 40% area fraction, and RANDOM (right) initial concentrations at $Pe = 10^4$. The velocity field is post-processed from the vesicle simulation with $AF = 40\%$ and $VC = 1$. See Fig. 3.11 for mixing with vesicles.*

3.4.4 Effects of Viscosity Contrast

Vesicles manifest different dynamics under simple 2D shear flow: either tank-treading or tumbling. Increasing viscosity contrast leads to a transition from tank-treading to tumbling motion [90]. In order to identify the effects of the viscosity contrast on mixing, we study mixing in vesicle suspensions with 5% area fraction and the viscosity contrasts of 1, 5 and 8. We run the simulations for the layer initial condition and demonstrate the mixing efficiency η with respect to time in Fig. 3.13. Figures 3.13(a), 3.13(b) and 3.13(c) show that an increase in the viscosity contrast results in additional mixing efficiency. However, for this initial condition, the viscosity contrast has less of an effect than the area fraction on the mixing efficiency.



(a) VC = 1 (b) VC = 5 (c) VC = 8

Figure 3.13: *Effects of viscosity contrast on mixing. We consider the layer initial condition and* 5% *area fraction of vesicles. The results show that the viscosity contrast has less of an effect on the mixing efficiency than the area fraction. We observe an increase in the maximum mixing efficiency* η *as the viscosity contrasts increases from 1 to 8 (see (a) and (c)), and this might stem from the fact that vesicles start to tumble for VC* ≥ 5.

To further investigate the effect of the viscosity contrast, we run the simulations for the dye initial condition. We consider 5% and 10% area fractions and the viscosity

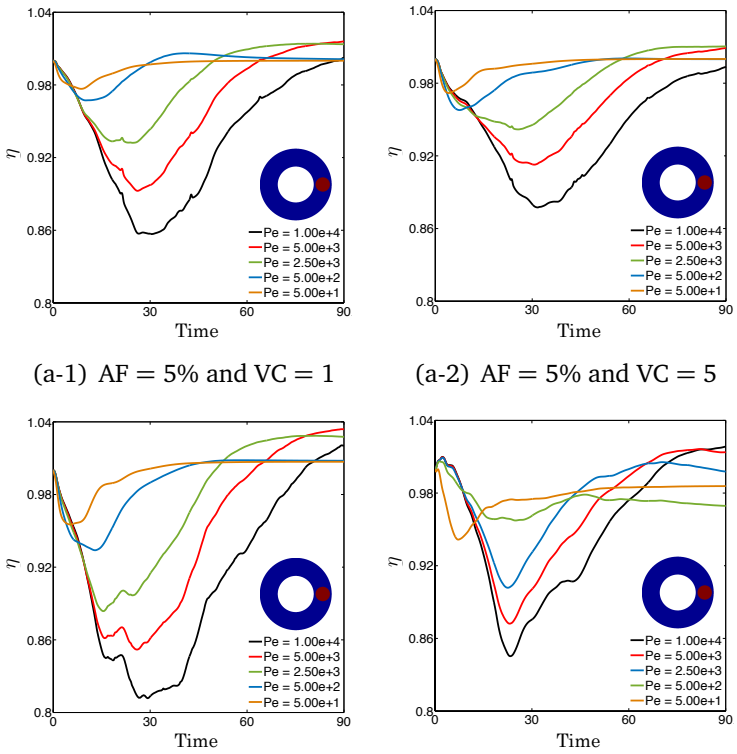
contrasts of 1 and 5. As we observed Fig. 3.10, mixing is suppressed in the presence of vesicles. The efficiency decreases further when the area fraction increases from 5% (Fig. 3.14(a-1)) to 10% (Fig. 3.14(b-1)) since more regions of the concentration will be trapped. Similar to what is observed for the layer initial condition, increasing in viscosity contrast from 1 to 5 promotes mixing for both area fractions, especially around time 30.

3.4.5 Summary

In an attempt to predict the mixing efficiency for given initial concentration ϕ_{IC}, we introduce a measure M of the ability of mixing for the default Couette flow

$$M = \frac{\int_\Omega |\nabla \phi_{\mathrm{IC}} \cdot \mathbf{v}| \, d\Omega}{\|\nabla \phi_{\mathrm{IC}}\|_{L^2} \|\mathbf{v}\|_{L^2}} \tag{3.11}$$

where $\nabla \phi_{\mathrm{IC}} = (\frac{\partial \phi_{\mathrm{IC}}}{\partial r}, \frac{1}{r}\frac{\partial \phi_{\mathrm{IC}}}{\partial \theta})$ and $\mathbf{v} = (v_r, v_\theta)$ is the velocity field of the default Couette flow. Equation (3.11) is a normalized L^1 norm of the advective term. We tabulate various initial conditions, the corresponding M values, and the minimum and the maximum efficiencies $\eta_{\min}$, $\eta_{\max}$ in Table 3.3. For all the initial concentrations that we consider except *LAYER*, M is initially non-zero meaning that mixing will occur due to advection in addition to diffusion. For these initial concentrations, a vesicle flow suppresses mixing by creating trapped regions. For the *LAYER* initial concentration, the advective term is initially zero and hence mixing occurs only due to diffusion. The vesicle flow provides better stirring and hence better mixing of this initial concentration than the default Couette flow. In order to verify this observation, we consider a slightly perturbed initial concentration in Fig. 3.15(c), which has a concentration gradient such that the advective term is initially non-zero. The M value for this initial concentration



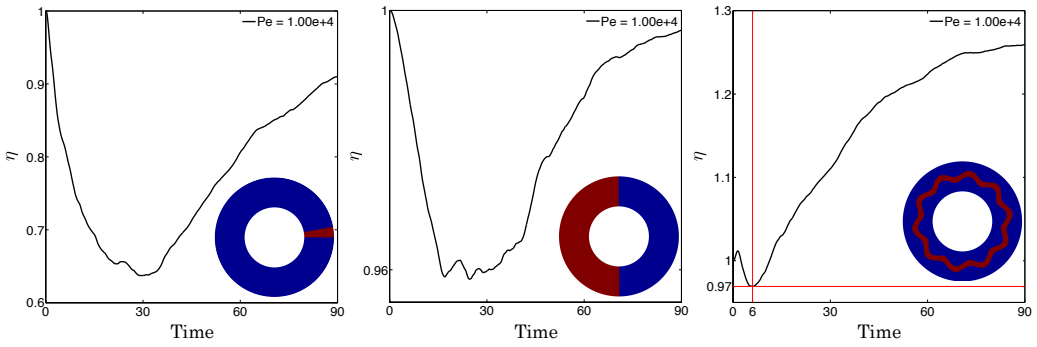
(a-1) AF = 5% and VC = 1 (a-2) AF = 5% and VC = 5

(b-1) AF = 10% and VC = 1 (b-2) AF = 10% and VC = 5

Figure 3.14: *Effects of the area fraction and viscosity contrast on mixing of the dye initial concentration. We consider the area fractions 5% (top) and 10% (bottom) and the viscosity contrasts 1 (left) and 5 (right). Unlike the layer initial condition, the presence of vesicles suppresses mixing of the dye initial concentration by creating trapped regions, i.e.* $\eta < 1$*. Increasing in area fraction from 5% to 10% decreases the mixing efficiency further. However, vesicles start to tumble for VC* ≥ 5*, which leads to more complex vesicle dynamics which promotes mixing. Therefore, the efficiency increases as viscosity contrast increases from 1 to 5, which is also observed for the layer initial condition in Fig. 3.11.*

is $M = 0.97$. We simulate mixing of this initial concentration with the vesicle flow at AF $= 40\%$ and VC $= 1$. The mixing efficiency η is shown in Fig. 3.15(c). For $t \in [0, 12]$, the vesicle flow suppresses mixing and the minimum efficiency is attained at around $t = 6$, but for $t \in [12, T_h]$ the vesicle flow promotes mixing. To explain this behavior we

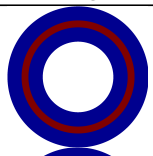
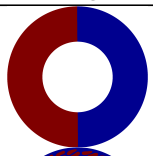
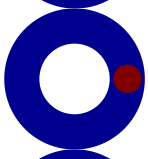
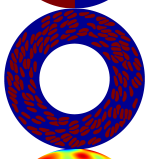
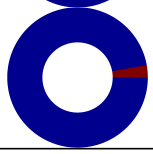
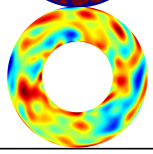
show frames from the mixing simulations with the vesicle flow and the default Couette flow in Fig. 3.16. Before $t = 6$, mixing occurs due to advection in addition to diffusion in the default Couette flow; however, after $t = 6$, the concentration field approaches the *LAYER* intial condition whose gradient only has a radial component while the velocity field does not have radial component. Therefore, the advective term becomes zero, and mixing is dominated by diffusion in the default Couette flow. Consequently, vesicle flows create trapped regions, which is responsible for a poor quality of mixing. Therefore, as long as the default Couette flow mixes the solute with both advection and diffusion, it provides better mixing quality than vesicle flows. However, in the cases where mixing occurs only due to diffusion in the default Couette flow (such as the layer initial condition), vesicle flows provide better mixing quality than the default Couette flow since the solute is mixed with advection in addition to diffusion and the transport is advection dominated.



(a) Mixing efficiency vs. time (b) Mixing efficiency vs. time (c) Mixing efficiency vs. time

Figure 3.15: *The mixing efficiency η as a function of time for Pe = 1E+4. The vesicle flow is the same for all initial conditions and has 40% area fraction and no viscosity contrast. We show frames from the mixing simulation of (c) in Fig. 3.16. The proposed measure* (3.11) *for this initial concentration is $M = 0.97$.*

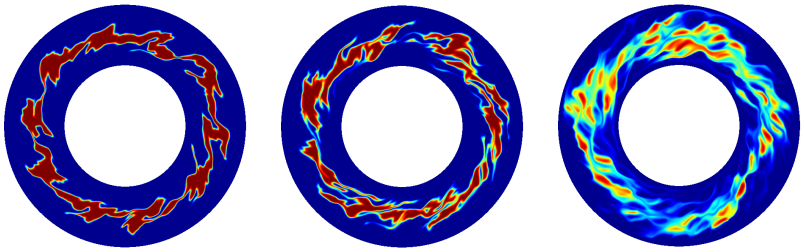
Table 3.3: *We report the proposed measure (3.11) for various initial concentrations ϕ_{IC} together with the minimum and maximum mixing efficiencies, $\eta_{\min}$ and $\eta_{\max}$, that they deliver with the vesicle flow of area fraction AF = 40% and viscosity contrast VC = 1 at Pe = 1E+4. Here, red is for $\phi_{IC} = 1$ and blue is for $\phi_{IC} = 0$. The mixing efficiencies of two additional initial concentration fields are in Fig. 3.15.*

ϕ_{IC}	M	$\eta_{\min}$	$\eta_{\max}$	ϕ_{IC}	M	$\eta_{\min}$	$\eta_{\max}$
	0	1	1.34		0.98	0.96	1
	0.52	0.63	1		1.86	0.98	1
	0.69	0.64	1		2.29	0.998	1

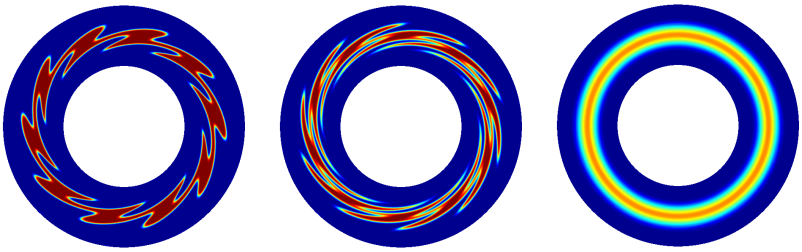
3.5 Conclusions

Using our integral equation solver and a pseudo-spectral advection-diffusion solver, we have studied mixing in vesicle suspensions. We have compared mixing in the absence and the presence of vesicles and investigated the effects of the Peclet number, area fraction, and viscosity contrast. The main outcomes are:

- For the same average Peclet number, the presence of vesicles suppresses mixing in most of the cases, and increasing the area fraction suppresses it more. However, there are special initial conditions of the passive scalar for which there is no advective mixing in the absence of vesicles. The presence of vesicles promotes mixing in those cases by providing advective mixing. In order to estimate whether the presence of vesicles promotes or suppresses mixing, we define a measure



(a-1) $t = 3$ (a-2) $t = 6$ (a-3) $t = 24$



(b-1) $t = 3$ (b-2) $t = 6$ (b-3) $t = 24$

Figure 3.16: *Frames from the mixing simulation of the initial condition in Fig. 3.15(c). The frames at the first row are from the simulation with the vesicle flow of 40% area fraction and no viscosity contrast. The ones at the second row are from the simulation with the default Couette flow. Both simulations have Pe = 1E+4. The corresponding mixing efficiency is in Fig. 3.15(c). In particular, the mixing efficiencies at the instances we show here are* $\eta(t = 3) = 0.99$, $\eta(t = 6) = 0.97$ *and* $\eta(t = 24) = 1.08$.

$M \propto \|\nabla\phi_{\text{IC}} \cdot \mathbf{v}\|_{L^1}$ where ϕ_{IC} is the initial concentration field and $\mathbf{v}$ is the default Couette velocity field. For M being zero mixing occurs due to diffusion in the absence of vesicles and hence vesicle flows promote mixing.

- For the same average Peclet number and the same area fraction, increasing viscosity contrast increases the mixing efficiency since the vesicles show more complicated dynamics at higher viscosity contrast, leading to a more complicated stirring of the solute.

Chapter 4

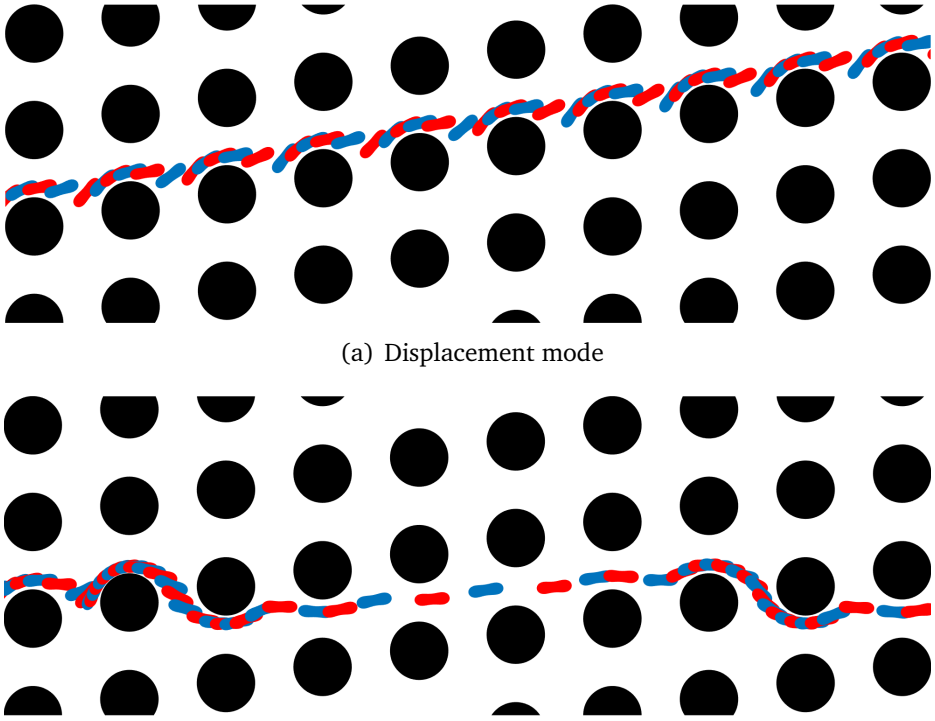
Sorting Same-Size Red Blood Cells in Deep Deterministic Lateral Displacement Devices

In this chapter[1] we present our computational study on deformability-based sorting of same-size red blood cells (RBCs) via deterministic lateral displacement (DLD). Microfluidic sorting of deformable particles finds many applications, for example, medical devices for cells. DLD is one of them. Particle sorting via DLD relies only on hydrodynamic forces. Our goal is to investigate if it is possible to separate deformable particles that have the same size but different mechanical properties and if so, quantitatively characterize the physical mechanisms that enable the sorting. We perform cell simulations using our integral equation solver for vesicle flows in two dimensions introduced in Chapter 2. It turns out cells moving with a sufficiently high positive inclination angle with respect to the flow direction displace laterally while those with smaller angles travel with the flow streamlines. Thereby, deformability-based cell sorting is possible. The underlying mechanism here is cell migration due to the cell's positive inclination and the curved flow lines. The higher the inclination is, the farther the cell can travel laterally. We also assess the efficiency of the technique for dense suspensions. Most of the cells in dense suspensions flow in the same direction no matter what their deforma-

[1] This chapter is based on work that has been published in [77]. The authors equally contributed.

bility is, which makes sorting difficult. Our study helps in the design and optimization of DLD devices.

4.1 Introduction



(a) Displacement mode

(b) Zig-zag mode

Figure 4.1: *Transport modes of red blood cells (RBCs) flowing in a DLD device. The repeated blue-red RBC shapes indicate different time snapshots from the trajectory of a single cell. The DLD device consists of arrays of pillars (filled black circles). The flow is from left to right. Each pillar column is slightly shifted vertically with respect to the previous column. Separation of cells using DLD depends on their orientation, elasticity and viscosity. Here, the RBC in (a) has a lower viscosity than the one in (b). The less viscous RBC moves along the inclination of pillars while the more viscous one moves in the flow direction. The former transport mode is called "displacement" and the latter is called "zig-zag". After multiple interactions between the RBCs and the pillars, the displacing RBC shows more lateral displacement than the zig-zagging one. Thereby, they are laterally separated. Here, the tilt angle of the pillar rows is* 0.17 rad.

Sorting biological cells by their mechanical properties (e.g., elasticity, cytoplasmic viscosity) is an important process in rapid medical diagnoses and tests using lab-on-a-chip technology. Therefore, microscale cell separation techniques have been of great interest. These techniques (see [13, 20] for an extensive review of them) take advantage of various cell fingerprints such as size, shape and deformability. For example, malaria-infected red blood cells and metastatic cancer cells can be differentiated based on their deformability as the malaria-infected cells are stiffer [159] and the metastatic cancer cells are softer than their healthy counterparts [56]. The microscale cell separation techniques are divided into two categories: active and passive. The former uses an external force field, e.g., electrophoresis is an active method using an electric field to separate cells of different sizes and charges. Passive separation relies mainly on the device geometry and on hydrodynamic interactions between particles and the device. Thus, passive techniques are cheap and readily available. Deterministic lateral displacement is a passive particle separation technique introduced by [71] to separate rigid particles from their dilute suspensions based on their sizes.

A DLD device consists of arrays of pillars (see Fig. 4.1). The pillar grid forms a lattice but the lattice vectors are not orthogonal. That is, the pillars are vertically aligned but their "horizontal" or "diagonal" alignment direction is at an angle with the x-axis, which is also the flow direction axis. This arrangement determines a critical particle size[34]. Particles larger than the critical size move along the direction defined by the pillars (see Fig. 4.1(a)) while those smaller than the critical size move with the flow (see Fig. 4.1(b)). The former transport mode is called "displacement" and the latter is called "zig-zag". After several particle-pillar interactions, the displacing

particle is separated laterally from the zig-zagging one because the latter has almost zero net lateral displacement. Notice that Fig. 4.1 shows snapshots from simulations of cells, however, these trajectories are characteristic of the transport modes and the same for the rigid particles as well. Analytical DLD theory has been developed for dilute suspensions of rigid spherical particles under the assumption that the particle-pillar interactions dominate the particle-particle interactions in determining the particle trajectories. Critical particle sizes are computed and the devices are designed based on this assumption [33, 34, 72]. However, dilution of suspensions requires a pre-treatment of the sample, which is time consuming and expensive and high volume fractions of suspensions are needed for high throughput. Therefore, the performance of DLD for dense suspensions is also of interest. Additionally, dynamics of non-spherical and deformable particles such as cells flowing in DLD devices needs to be investigated in order to design DLD devices for sorting those particles. So, although the DLD technique has been frequently used and is promising, several questions regarding its performance remain open. Given the wide spectrum of possible applications, here we focus on the sorting of human red blood cells.

Human blood consists of plasma and mainly white and red blood cells and platelets. White blood cells (WBCs) are mostly spherical with a diameter in the range 5μm - 20μm, red blood cells are biconcave with a diameter of 8μm and a thickness of 3μm and platelets are nearly rigid discoids (when they are not activated) with a diameter between 1μm - 3μm [133]. DLD has been used for fractionation of these components of human blood based on their sizes [34] and also used for separation of WBCs [34], RBCs [177], parasites [67] and circulating tumor cells [88, 108]. Diseases

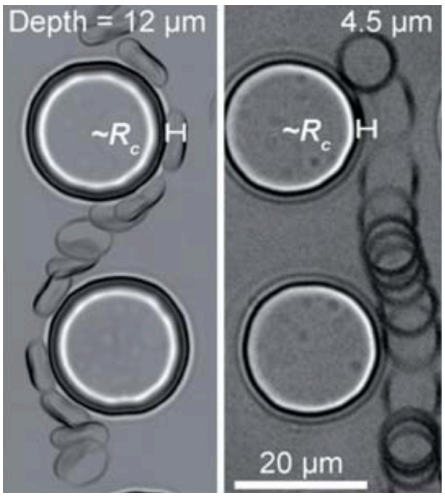


Figure 4.2: *RBC orientations in a deep (on the left) and a shallow (on the right) DLD devices (the images show top view and are taken from [19]). The heights of the pillars are $H = 12\mu m$ in the deep device and $H = 4.5\mu m$ in the shallow device. These are superimposed images of RBCs flowing in actual DLD devices. The fluid flows from top to bottom. In the deep device the RBC is orientated in a way that its thickness becomes its effective size. In the shallow device the confining walls in the out-of-plane direction pushes the RBC to move parallel to the walls. Therefore what we see is a disc and the effective size is its diameter. The deep DLD devices are preferable to shallow ones in practice since they reduce the risk of clogging and provide higher throughput. We only consider deep devices in our study.*

such as sickle cell anemia [15], diabetes [24], and malaria are responsible for changes in RBCs' deformability by altering both the cytoplasmic viscosity and the membrane's viscoelastic properties. Therefore, deformability-based sorting of RBCs could potentially identify and separate abnormal cells from blood. These considerations motivate our study of DLD devices. The parameters we used in this study are the membrane elasticity and cytoplasmic viscosity of RBCs, which determine their deformability.

A DLD device for RBC separation can be either *shallow* or *deep* Fig. 4.2. A shallow device has short pillars with a height less than the RBC diameter ($H \approx 4-5\mu$m) while a deep one has tall pillars with a height much greater than the RBC diameter

($H \gg D_{\mathrm{RBC}}$). RBCs orient differently in these devices. A shallow device pushes cells parallel to the confining walls in the out-of-plane direction and results in an effective size (when they are not deformed) equal to the cell diameter. In a deep device they orient themselves in such a way that their effective size becomes their smallest size, i.e., their thickness. In shallow devices the separation of cells is similar to that of rigid spherical particles. While a suspension of stiff and soft cells flows through a shallow device, soft cells deform more than stiff cells and hence their effective size becomes smaller than that of the stiff cells. This results in zig-zagging soft cells and displacing stiff cells, and hence separation. However, throughput is limited in shallow devices. In contrast, deep devices allow more cells along the pillars and hence higher throughput. Cells can show richer dynamics in deep devices than shallow devices since deep devices do not confine cells in the out-of-plane direction as much as shallow ones do. Therefore, the separation in deep devices does not occur by the same means as in shallow devices. It depends on complex dynamics of cells such as moving with a stationary angular orientation (i.e., tank-treading) or varying angular orientation (i.e.,tumbling) [65]. RBCs show this complex dynamics in shear and Poiseuille flows depending on the cytoplasmic viscosity and membrane elasticity [40, 117–119, 132]. Cell dynamics in these fundamental flows has been studied and is well-understood. However, the DLD flow is more complicated than these flows. How do such complex behaviors appear in deep DLD devices? How does sorting depend on them? How can we quantify them and explain sorting? These are the main questions we aim to address in this study.

Methodology. The main components of our approach is the mathematical model for the RBCs, the formulation and discretization, and the DLD device set-up. Following [119, 124], we model an RBC as an inextensible vesicle with a reduced area of 0.65. RBCs and vesicles share similar dynamical properties and the differences are minor in two dimensions. Since the Reynolds number is $\mathcal{O}(10^{-3})$ (see [115] for a summary of the experiments reported in the literature), we use the quasi-static Stokes approximation scheme presented in Section 2.2 and the numerical scheme in Section 2.3. Our DLD model has pillars with a circular cross-section only. Our two-dimensional (2D) model can represent deep DLD devices not the shallow ones because RBCs with a reduced area of 0.65 in the 2D model have the same orientation as RBCs in real deep devices (see Figures 4.1 and 4.2). Additionally, our 2D model does not include the wall effects and the cell-cell interactions in the out-of-plane direction which are negligible in deep devices but important in shallow ones. Unlike other numerical studies (discussed below) we include multiple rows and columns of pillars in our simulation domain (this results in $\mathcal{O}(100)$ pillars). We consider only one free parameter for the DLD device: the row-shift fraction ϵ (i.e., the angular orientation of the pillar orientation). We quantify the RBC's elasticity and viscosity with two dimensionless numbers: the capillary number C_a and viscosity contrast ν, respectively. First, we perform several simulations of RBCs with no viscosity contrast by changing the capillary number C_a in a DLD with the row-shift fraction $\epsilon = 0.1667$. Second, in the same device we vary the viscosity contrast ν and fix the capillary number to $C_a = 3.41$, which corresponds to a value for the flow of a healthy RBC through DLD with an average velocity of 1mm/s. This flow speed is in the range [1μm/s, 10mm/s] used in experiments on sorting RBCs

depending on their deformability [115]. Additionally, it is not so high that cells do not deform significantly and hence we can observe the effects of viscosity contrast on cell dynamics. Last, we vary the viscosity contrast and capillary number for fixed row-shift fractions $\epsilon = 0.1$ and $\epsilon = 0.1667$. Then we map out the parameter space for the transport modes as a function of C_a, ν, ϵ. We also perform simulations of dense suspensions in DLD and quantify the efficiency of the technique for dense suspensions. Summary of our findings:

- Relation to cell migration: cell migration is a phenomenon observed in a shear and confined Poiseuille flow. That is, cells having positive inclination with respect to the flow direction migrate towards low shear rate regions due to the curved flow lines. In our numerical studies, we observe that more deformable cells have higher positive inclination angles with respect to the flow direction than the less deformable ones (see Section 4.4.2). Since the velocity profile near the pillars is parabolic, the more deformable cells migrate further away from the pillars than the less deformable ones. Thus, the more deformable cells displace while the less deformable ones zig-zag.

- Effects of complex geometry: although we have discovered that cell dynamics in DLD gaps is similar to those in confined Poiseuille flows (see Section 4.4.2), the dynamics in the whole device is more complicated than the channel flow because a cell periodically moves from a confined gap to a less confined region between gaps (By "DLD gap", we refer to the flow space between two vertically aligned pillars.) So it is not possible to estimate the cell dynamics in DLD from a simpler flow such as a channel flow.

- Quantification of migration: we have computed the cell vertical migration velocity in DLD and demonstrated that it correlates well with the inclination angle (see Section 4.4.2).

- Pseudo-lift: By computing the so-called pseudo-lift (a measure of the alignment of local forces to the migration direction) acting for several cells, we have found 5× stronger pseudo-lift on cells with high inclination angles compared to cells with smaller inclination angles which are under either a negative or a weak pseudo-lift (see Section 4.4.3).

- Dense suspensions: finally, when we assessed the DLD efficiency for dense RBC suspensions, we have observed that most cells zig-zag in dense suspensions no matter what their capillary numbers and viscosity contrasts are. So it is difficult to separate small rigid particles or stiff cells from dense suspensions because these would zig-zag, too. This result agrees with the numerical [166] and experimental [73] studies which use shallow devices (see Section 4.3.4).

Contributions. To the best of our knowledge, this is one of the first studies investigating the effects of the complex RBC dynamics in deep DLD devices. We show that deformability-based sorting of RBCs is possible. We list our contributions below.

- We investigate the cell dynamics in DLD flow, i.e., we study the inclination angle and lateral velocity of the cells in Sections 4.3.1 and 4.3.2.

- We compare these cell-in-DLD dynamics with simple flows such as a free shear and confined Poiseuille flows in Sections 4.4.1 and 4.4.2.

- In order to quantify migration, we compute a new quantity in Section 4.4.3, which we call the "pseudo-lift", that turns out to be an indicative measure of migration.

- Lastly, we present phase diagrams for the transport modes in Section 4.3.3 and investigate the separation in dense RBC suspensions in Section 4.3.4 for different viscosity contrasts, capillary numbers and device configurations. The efficiency of DLD for dense suspensions in deep devices has not been studied experimentally or numerically. However, it is essential to investigate the dense suspension regime because the deep devices with dense suspensions provide higher throughput than the shallow devices with dilute suspensions.

Limitations. Our modelling approaches have several limitations. First, our simulations are in two dimensions. We have opted to use two-dimensional simulations because three-dimensional simulations of a DLD device including multiple pillars can be quite expensive for this purpose [151]. Second, our numerical scheme is valid only for Newtonian flows with no inertia (zero Reynolds number) and does not permit viscoelastic fluids in the interior and the exterior of the RBCs. Finally, our RBC model does not have cytoskeleton. Despite the limitations of our model, it reproduces actual experimental results without any tuning parameters. For example, it reproduces the phase diagram for size-based sorting of rigid spherical particles. Our model allows to explore the underlying DLD physics and explore the parameter space. However, our simulations do not shed light on several other factors affecting cell sorting. One of them is the effects of changes in the resting size and shape of an RBC. When an RBC

becomes diseased, not only its elasticity and viscosity change but its resting size. The other one is the effect (on cell sorting) of the lateral confinement in the DLD gaps. In our study, the confinement in the gaps does not allow a cell to tumble and leads it to move always with a positive inclination angle. In the case of a weaker confinement, the cell can tumble for high viscosity contrasts (i.e., for less deformable cases). We expect tumbling to increase the cell's effective size in those cases and hence result in displacing cells. For example, Ranjan et al. [149], Zeming et al. [177] observed that pillars with protrusions cause cells to tumble and tumbling cells start displacing for the same reason. Finally, we do not consider the effects of adhesion on cell sorting, which could be important for dense RBC suspensions. Blood suspensions are usually diluted in real applications [115]. In dilute suspensions the distance between the cell and the pillars is much greater than the adhesion length scale. Hence, we do not expect any adhesion effects on our results for a single cell flowing through a DLD device.

Related work. The study in [65] is very similar to us and the findings are consistent. They study both deep and shallow devices by combining simulations (3D smoothed dissipative particle dynamics) and experiments. They only consider three viscosity contrast values (0.25, 1, 5), whereas we study a much wider range of viscosity contrasts (up to 100). Also, we vary the capillary number (they do not). Going beyond the work in [65], we conduct a systematic study and map out the parameter space for the transport modes of cells. We investigate the cell dynamics in DLD in detail and compare those dynamics with simpler flows such as confined Poiseuille flow. We claim that cell migration is responsible for cell separation in DLD and quantify migration

with migration velocity and pseudo-lift. [65] shows our 2D simulations can capture the actual experiments and support our findings and rationale. In addition to [65] mentioned above, there have been a few numerical studies investigating the separation of deformable particles using DLD:

- Quek et al. [145] and Ye et al. [175] performed two-dimensional simulations of spherical deformable particles with no viscosity contrast in order to explain the effects of particle deformability on the separation,
- Krüger et al. [97] systematically investigated the effects of the capillary number C_a on the separation of RBCs with a constant viscosity contrast of $\nu = 5$ in shallow devices using three-dimensional simulations,
- Zhang et al. [179] studied circular, square, diamond and triangular pillar shapes for the separation of rigid particles and RBCs using two-dimensional simulations.

Overall, none of these studies have systematically investigated the separation of RBCs as a function of the capillary number and viscosity contrast in deep devices.

Regarding numerical methods, all the studies above used either the immersed boundary method [97, 145, 166], lattice-Boltzmann method [97, 166], fictitious domain method [175] or dissipative particle dynamics [65, 179]. Here, we use our algorithms based on the boundary integral equation formulation for Stokesian particulate flows. Additionally, most numerical studies of DLD mentioned in the previous paragraph reduced the simulation domain to a single pillar and used periodic boundary conditions. Since actual DLD devices have walls that result in zero net lateral flow,

an artificial force in the lateral direction needs to be introduced to mimic the wall effects in the periodic models [32]. Our model contains multiple pillars in the lateral and flow directions, and also the top and bottom walls. Therefore, we do not need to add any force mimicking the wall effects. The only parameters in our scheme are the physical parameters (device geometry, number of cells, viscosity contrast and capillary number) as well as the spatial discretization size. We use an adaptive, semi-implicit time-stepping scheme. No other (non-physical) parameters are necessary. We only have five free parameters in our model (other than the shape of the device): the time-step error tolerance, the points per cell, the discretization size of the rigid walls, the viscosity contrast and the capillary number. We have also performed a convergence study to verify our method in the DLD setting in Section 4.2.2.

There is only one numerical study [166] where simulations of dense RBC suspensions with different volume fractions were performed in shallow devices. The authors found out that the displacement mode breaks down as the volume fraction increases and most of the RBCs zig-zag independently. We also studied this by performing simulations of dense RBC suspensions with two different capillary numbers and area fractions in deep devices. We reached the same conclusions. In addition, we consider dense RBC suspensions with different viscosity contrasts. Our results show that, again, the displacement mode breaks down.

Organization of the Chapter. We explain the DLD theory in detail, our DLD model and its validation in Section 4.2. We present our results in Section 4.3. Then, in Section 4.4 we quantify the cell dynamics in DLD and compare them with those in shear and

confined Poiseuille flows. Finally, we summarize our results in Section 4.5.

Notation. We summarize the main notation used in this chapter in Table 4.1.

Table 4.1: *List of frequently used notation.*

Symbol	Definition
C_a	Capillary number (4.3), dimensionless RBC elasticity
ν	Viscosity contrast (4.4), dimensionless cytoplasmic viscosity
ϵ	Row-shift fraction: tangent of the tilt angle of the pillar rows
x	Main flow direction
y	Lateral direction, in which separation occurs

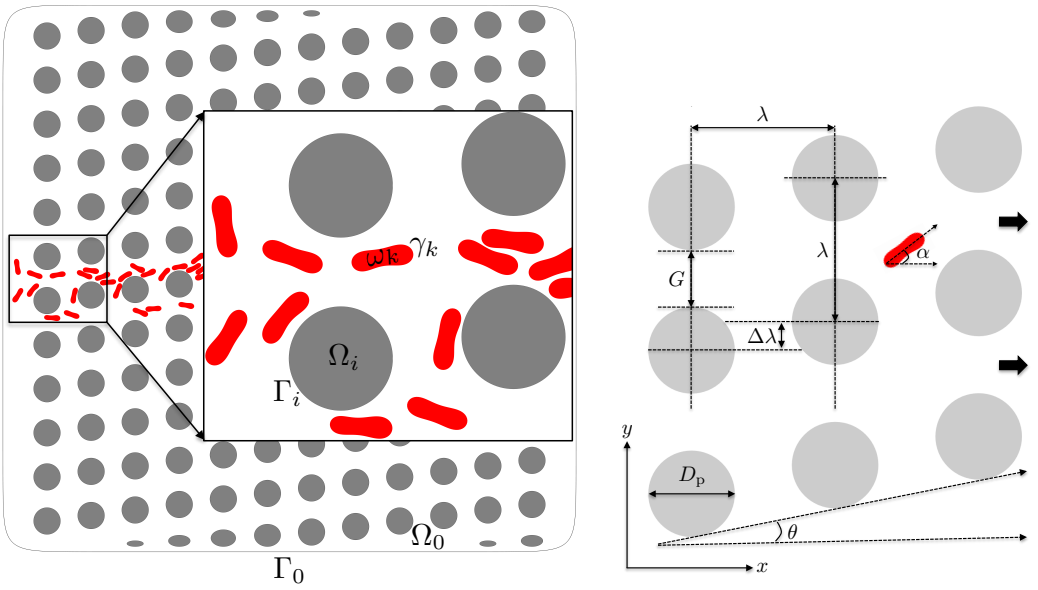
4.2 Numerical Model

We use our scheme based on the integral equation formulations in Section 2.3. Here, we introduce our DLD model in Section 4.2.1. We, then, verify and validate our model in Sections 4.2.2 and 4.2.3, respectively. Lastly, we discuss the dimensionless numbers used in this study in Section 4.2.4.

4.2.1 DLD Model

A DLD device of circular pillars can be uniquely determined by three parameters: the diameter of a pillar D_{p}, the center-to-center distance between two neighboring pillars λ and the row shift $\Delta\lambda$. See Fig. 4.3(b) for the DLD geometry.

Geometry. Our DLD device consists of circular pillars all of which have the same diameter $D_{\text{p}} = 15\mu\text{m}$. The fluid flows in the x direction. The center-to-center distance



(a) An RBC suspension in a DLD device

(b) DLD parameters

Figure 4.3: *We illustrate the domain of a cell suspension in a DLD device in (a). The interior and boundary of the i^{th} pillar are denoted by Ω_i and Γ_i (Ω_0 and Γ_0 are the ones of the exterior wall). The empty circles are the pillars in the left and the right imaginary columns. The walls pass through these pillars' centers. The blue arrows show the parabolic velocity* $\mathbf{U}$ *imposed as a boundary condition in the gaps Γ_0^g between the imaginary pillars. In (b) we show the DLD parameters where x is the flow direction. The geometry is uniquely defined by the pillar diameter D_p, the center-to-center distance λ, and the row shift $\Delta\lambda$. Using these parameters we can determine the gap size G, the row shift fraction $\epsilon = \Delta\lambda/\lambda$ and the tilt angle of the pillars (the angle between the flow direction and the alignment direction of the pillars) $\theta = \tan^{-1}(\epsilon)$. The red cell in (b) has a positive inclination angle α with respect to the flow direction, which is defined in Section 2.5.1.*

between neighboring pillars is $\lambda = 25\mu$m and the same in both directions. This results in a gap size of $G = \lambda - D_{\text{p}} = 10\mu$m. Pillars centered at the same x coordinate form a "column". Each pillar in a neighboring column is shifted by $\Delta\lambda$ in the y-direction with respect to the previous one, which defines the row shift fraction $\epsilon = \Delta\lambda/\lambda$. Thus, the "rows" are tilted with an angle $\theta = \tan^{-1}(\epsilon)$. We define "lane" to be a path between two diagonally parallel rows.

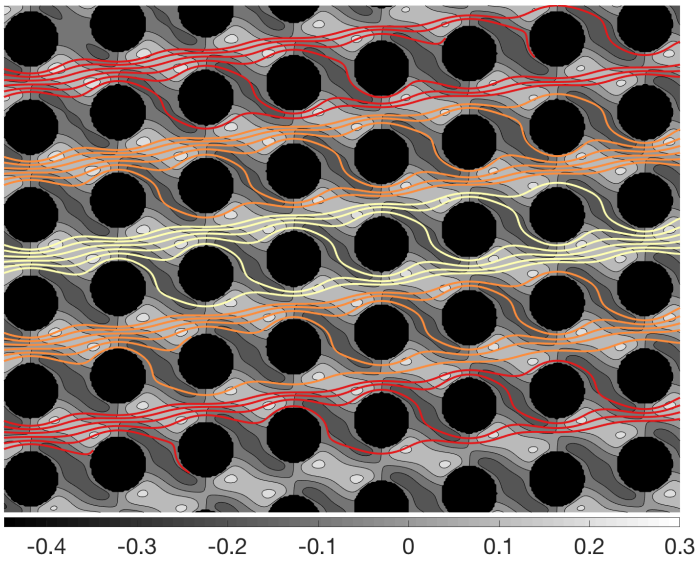


Figure 4.4: *Velocity in the vertical direction (represented by gray-scale contour colors) and streamlines in the absence of particles are superimposed. The row-shift fraction is* $\epsilon = 0.1667$*, which corresponds to the period* $n_p = 6$*. The flow is from left to right. See Fig. 4.8 for the velocity magnitude in the whole device and the velocity field in a gap. Each streamline interacts with a pillar, then swaps a lane. The DLD theory assumes that rigid particles do not distort the streamlines significantly. Depending on their size, they either follow these streamlines in zig-zag mode or cross the streamlines and stay in the same lane in displacement mode.*

DLD theory for rigid spherical particles. The row-shift fraction ϵ sets what is referred to as the period p of the device. "Period" sets a length scale in which the column arrangement is exactly repeated. Therefore, we can divide the unperturbed flow in a gap (i.e., in the absence of any particles) into n_p flow streams of equal mass flux. If we assume that the unperturbed flow does not change significantly when particles flow, the width of the stream adjacent to a pillar becomes the critical particle size [34]. Particles small enough to fit into one of the streams stay in their streams by zig- zagging, whereas the larger ones cannot. As the tilt angle θ reduces (i.e., the row-shift fraction ϵ reduces or the period n_p increases), the stream width decreases and hence

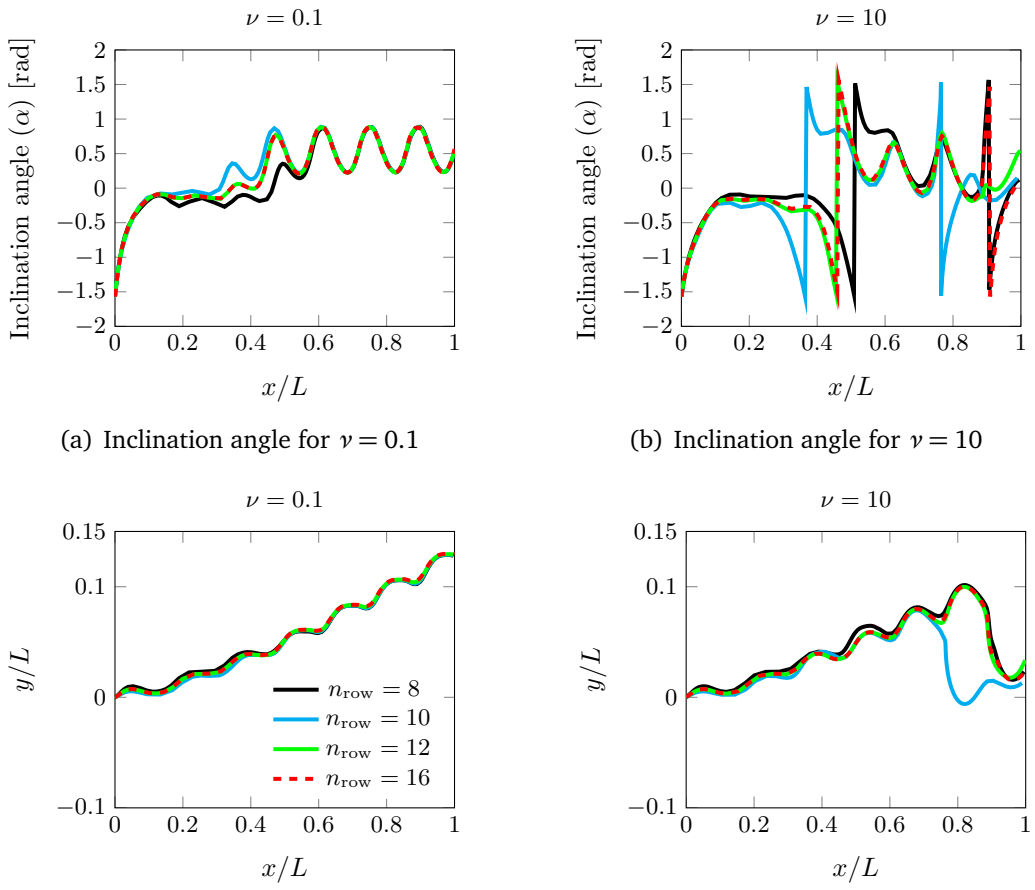
the critical particle size decreases. This theory is in good agreement with experiments with spherical rigid particles [33, 34].

Simulation domain. DLD devices usually consist of $\mathcal{O}(10)$ rows and $\mathcal{O}(100)$ columns, which results in $\mathcal{O}(1000)$ pillars in a device. Performing simulations of cell separation in a whole DLD device is computationally expensive [151]. This renders its systematic analysis infeasible even in 2D. In order to evade the computational cost the numerical studies conducted so far in this area reduced the simulation domain to a single pillar by assuming periodicity [97, 145, 166, 175, 179]. However, one pillar with periodic conditions requires imposing an artificial lateral force to enforce zero net lateral flow since in real DLD devices the lateral flow is restricted by the walls.

Due to the high computational costs, most of our numerical experiments are conducted using only one period, i.e., at least $\lceil 1/\epsilon \rceil$ pillars in the flow direction instead of using just one pillar. We confine the pillar lattice with an exterior wall to enforce zero net lateral flow. This raises the questions as to whether the wall effects introduce large errors. We numerically determined that the errors are negligible if we use 12 pillars in the y direction (i.e., 12 rows) and $\lceil 1.5(1/\epsilon) \rceil$ pillars in the x direction (i.e., $\lceil 1.5(1/\epsilon) \rceil$ columns) (see Fig. 4.3(a) and Fig. 4.8 as an example). We found the sufficient amount of rows and columns that must be included in our domain to minimize the wall effects as follows. We considered a DLD device with the row-shift fraction $\epsilon = 0.1667$, i.e., $n_p = 6$. We performed simulations of RBCs with the capillary number $C_a = 0.648$ and two different viscosity contrasts $\nu = 0.1$ and $\nu = 10$. The RBC with the lower viscosity contrast displaces and the one with the higher viscosity contrast zig-zags.

We, first, fixed the number of columns to $n_{\text{col}} = \lceil 1.5p \rceil$ and varied the number of rows $n_{\text{row}} = (8, 10, 12, 16)$. We present the inclination angles and trajectories of the RBCs during their motions in Fig. 4.5. The displacing RBC ($\nu = 0.1$) shows different transitions in the devices with $n_{\text{row}} = (8, 10)$ than those with $n_{\text{row}} = (12, 16)$. But the RBC has the same trajectories and inclination angles after the transition for all n_{row}. The zig-zagging RBC ($\nu = 10$) follows similar trajectories with similar variations in the inclination angles in the devices with $n_{\text{row}} = (12, 16)$. Since the trajectories and inclination angles do not change much after $n_{\text{row}} \geq 12$, we decided to include $n_{\text{row}} = 12$ rows in our DLD domain.

Dynamics of RBC given by a simulation in a single period of a DLD device must repeat itself as the simulation domain includes more periods if the simulation in a single period is not spoiled by the side wall effects. In order to determine if our simulations in a single period is accurate, we fixed the number of rows to $n_{\text{row}} = 12$ and considered the number of columns $n_{\text{col}} = (p, 2p)$ for the same examples above. We present the results in Fig. 4.6. The displacing RBC goes through a transition within one period and follows a trajectory along the pillars. As one more period is included in the domain, the trends in neither the inclination angle nor the trajectory change. The zig-zagging RBC falls down a lower lane once within one period and repeats this motion if the device includes one more period. However, the RBC flips near the end of the first period in the device with $n_{\text{col}} = p$ columns (see Fig. 4.6(b)). The RBC also flips in the device with two periods but near the end of the second period not the first one. This can be avoided by having $n_{\text{col}} = \lceil 1.5p \rceil$ columns in the domain. It seems that it suffices to have $n_{\text{col}} = \lceil 1.5p \rceil$ number of columns in our domain and perform



(a) Inclination angle for $\nu = 0.1$

(b) Inclination angle for $\nu = 10$

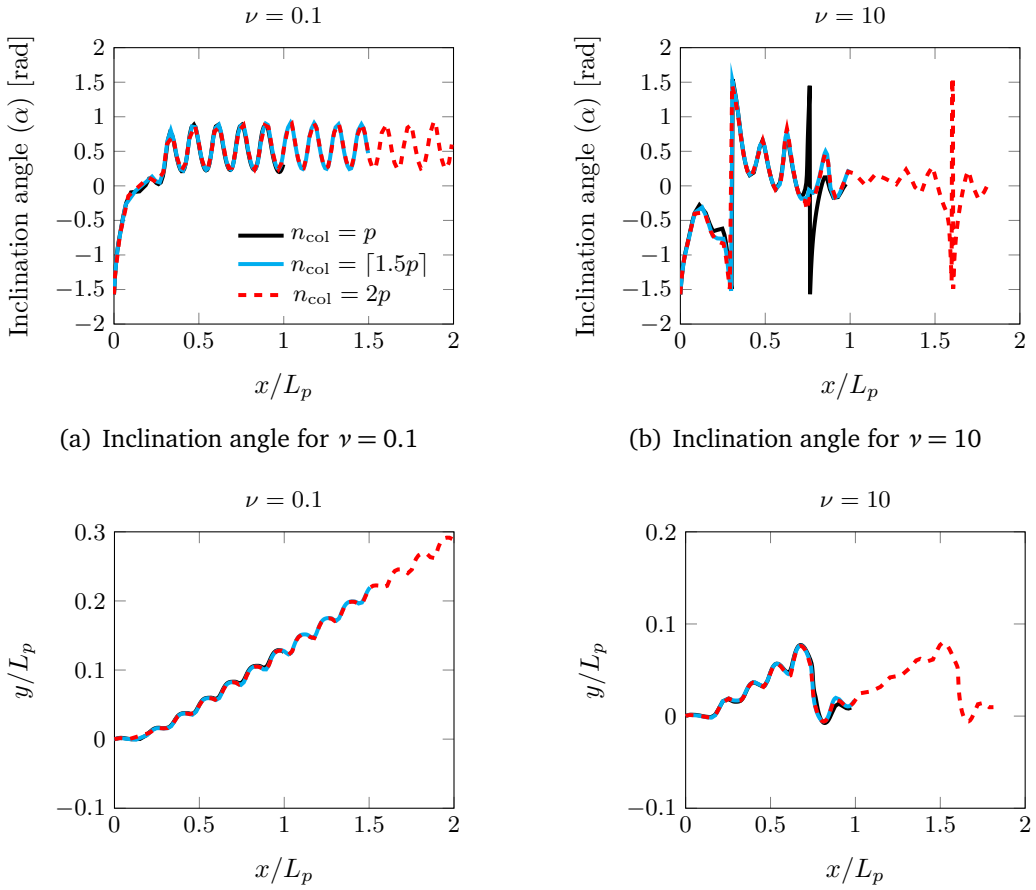
(c) Trajectory for $\nu = 0.1$

(d) Trajectory for $\nu = 10$

Figure 4.5: *Effects of the top and bottom walls on the RBC behavior. As increasing the number of rows n_{row} we plotted the RBCs' inclination angles and trajectories. While the trajectories and inclination angles are evidently different for the number of rows $n_{\text{row}} = 8$ and $n_{\text{row}} = 10$, there is no significant difference between the results given by $n_{\text{row}} = 12$ and $n_{\text{row}} = 16$. Therefore, we decided to include $n_{\text{row}} = 12$ number of rows in our domain. x and y coordinates are normalized by the length of the device L.*

simulations within one period to alleviate the wall effects.

We also investigated whether our DLD model is sensitive to the initial lateral position and inclination angle of an RBC. We performed several simulations with RBCs



(a) Inclination angle for $\nu = 0.1$

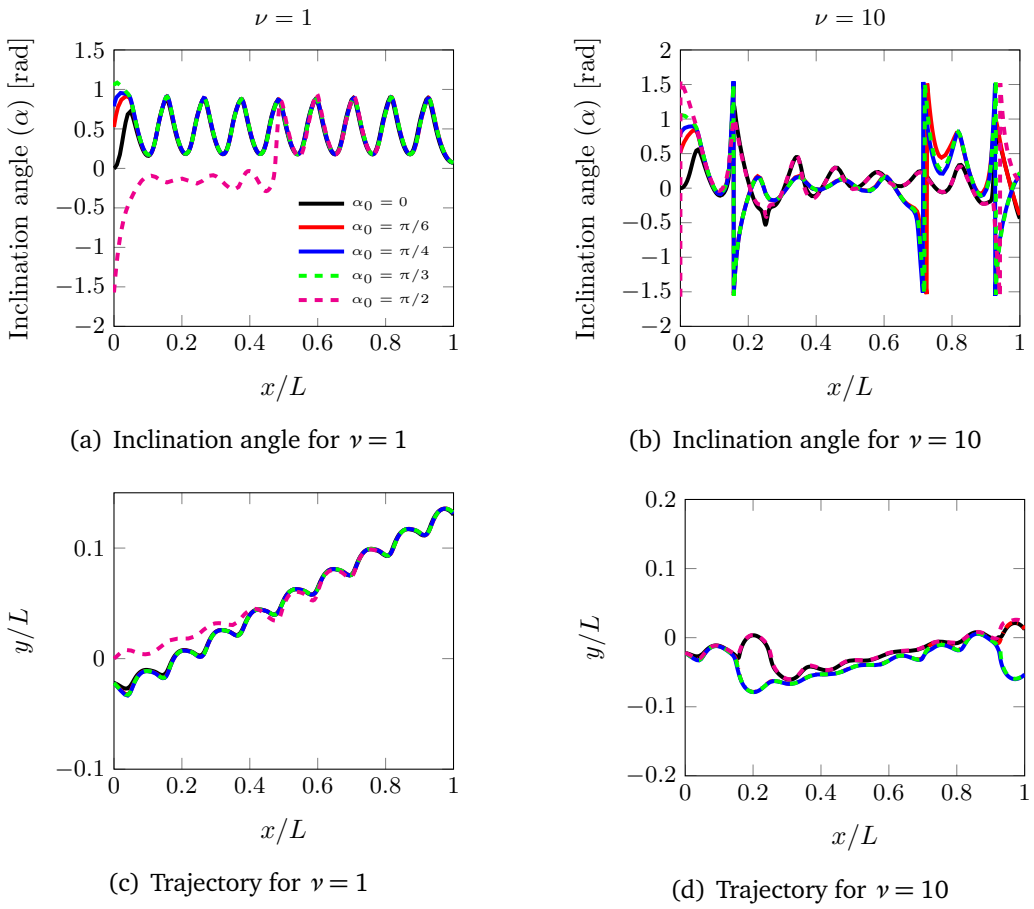
(b) Inclination angle for $\nu = 10$

(c) Trajectory for $\nu = 0.1$

(d) Trajectory for $\nu = 10$

Figure 4.6: *Effects of the side walls on RBC dynamics. We considered the same RBCs in Fig. 4.5. The DLD device also has the same period $n_p = 6$. We fixed the number of rows to $n_{\text{row}} = 12$ and performed the simulations in a single period ($n_{\text{col}} = p$) and two periods ($n_{\text{col}} = 2p$) of the device. We plotted RBCs' inclination angles and trajectories. We found that the RBC shows the same dynamics in a single period in both devices with $n_{\text{col}} = \lceil 1.5p \rceil$ and $n_{\text{col}} = 2p$. x and y coordinates are normalized by the length of one period of the device L_p.*

initialized at two different lateral positions and with various inclination angles. We chose a DLD device with the row-shift fraction $\epsilon = 0.1667$ and considered the viscosity contrasts $\nu = (1, 10)$. In Fig. 4.7 we show the trajectories and inclination angles (α) of



(a) Inclination angle for $\nu = 1$

(b) Inclination angle for $\nu = 10$

(c) Trajectory for $\nu = 1$

(d) Trajectory for $\nu = 10$

Figure 4.7: *Angular orientations and trajectories of RBCs of viscosity contrasts $\nu = 1$ and $\nu = 10$ initialized with different inclination angles and at lateral positions. Under these conditions the less viscous displaces and the more viscous one zig-zags. (a,b) Inclination angles α as the RBC moves along the flow direction x; (c,d) positions of the RBCs' centroids. x and y coordinates are normalized by the length of a period L. The scales for x and y coordinates are chosen differently for better visualization.*

the RBCs in those simulations. The less viscous RBC displaces (on the left in Fig. 4.7) while the more viscous one zig-zags (on the right). For all initial inclination angles α_0 the displacing RBCs reach an equilibrium angular orientation alternating between two

angles (see Fig. 4.7(a)). They also follow the same trajectories after different transients (see Fig. 4.7(c)). Additionally, the RBC with $\alpha_0 = \pi/2$ is initialized at a different lateral position and it also attains the equilibrium orientation and the trajectory. In the zig-zag mode there is no equilibrium in dynamics. Although trajectories are sensitive to the RBC's initial position and inclination angle [145], the transport mode is supposed to persist. For the zig-zagging case (high viscosity contrast) we observe different trends in the inclination angles in Fig. 4.7(b) and trajectories in Fig. 4.7(d) depending on the initialization. The RBCs with $\alpha_0 = (0, \pi/2)$ and those with $\alpha = (\pi/6, \pi/4, \pi/3)$ show the same variations in the inclination angles and trajectories. Ultimately all of the more viscous RBCs zig-zags. Therefore, we can conclude that our model is insensitive to how an RBC is initialized. The initial locations of the cells are in the middle lane. Since the pillars are shifted laterally, we end up with pillars crossing the exterior wall at the top and the bottom. We found out that if we just remove those pillars, the empty spaces result in less hydraulic resistance and hence induce a lateral pressure gradient. This breaks the homogeneity of the flow and introduces significant errors in the cell trajectories [98, 167]. Therefore, we decided to replace the circular pillars crossing the walls with elliptical pillars in such a way that they maintain the same vertical spacing with the neighboring pillars. Those non-circular pillars near the top and bottom walls provided a homogeneous flow in the middle region in our model (see Fig. 4.8 for our DLD model and the unperturbed velocity for $n_p = 6$). Henry et al. [65] showed that RBCs might have additional zig-zag modes unlike rigid particles, such as zig-zagging within a period or zig-zagging that requires more than a period to take place. In order to capture those zig-zagging modes, we perform simulations of a cell by initializing

it at several lateral positions. This is equivalent to simulating the cell for more than one period because in each simulation the cell confronts the first pillar at a different lateral position. We label the cell zig-zagging if it zig-zags in any of these simulations and displacing if it displaces in all of the simulations.

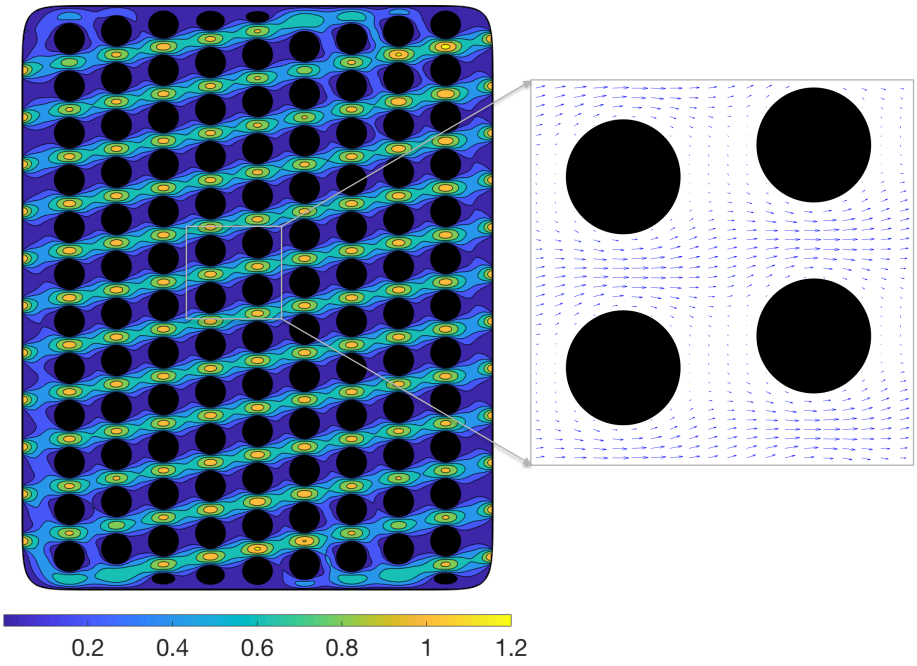


Figure 4.8: *Velocity magnitude (on the left) and velocity in a gap (on the right) in our DLD model for $\epsilon = 0.1667$, i.e., $n_p = 6$. Our DLD model consists of 12 rows and $\lceil 1.5n_p \rceil$ columns of pillars (i.e., 9 columns here). We draw the side walls such that they pass through the centers of the shifted imaginary columns (not shown). Then we impose a parabolic velocity profile (4.1) on the boundaries which correspond to the gaps between the imaginary pillars as a boundary condition. We replaced the circular pillars crossing the top and the bottom walls with the elliptical ones. That leads to a homogeneous velocity in the device. See Fig. 4.4 for the velocity in the y-direction and the streamlines for this DLD configuration.*

Boundary conditions. We want to reflect the periodicity of the DLD device in our model by imposing the velocity as a boundary condition at the intake and the outtake.

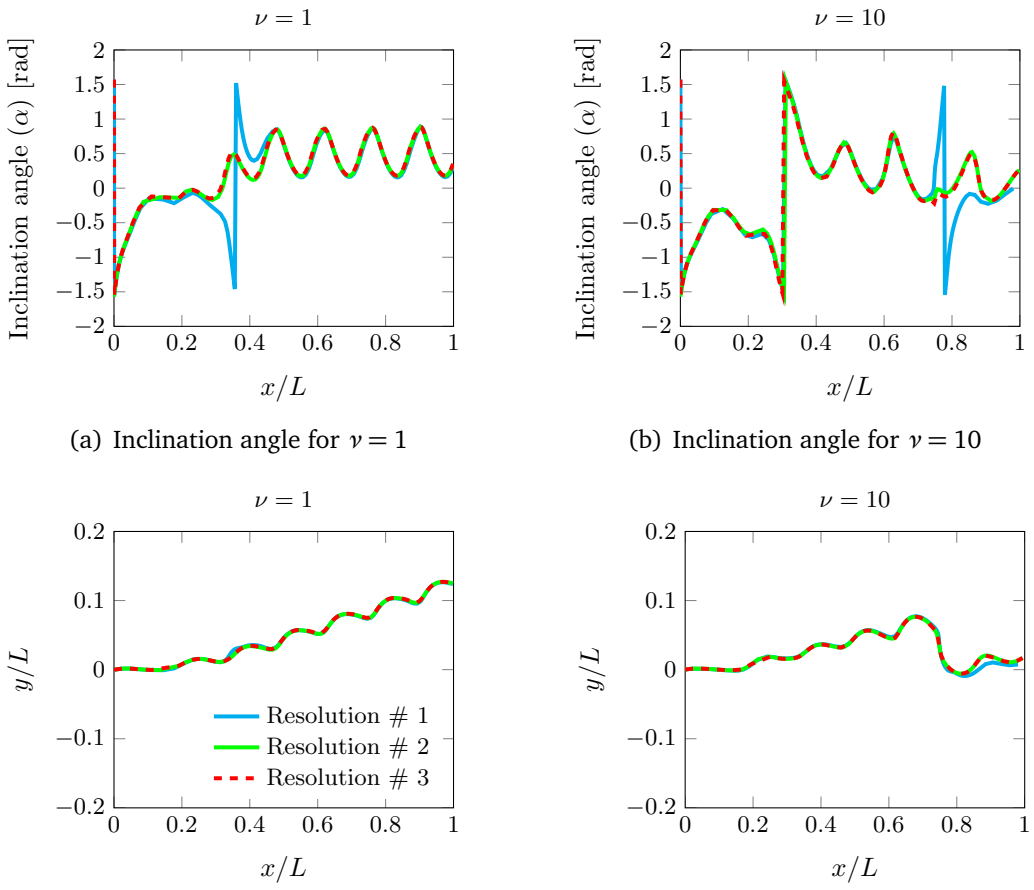
For this purpose, we place the side walls as if they pass through the imaginary columns of the pillars shifted down on the left and up on the right by $\Delta\lambda$ (the empty circles in the left figure in Fig. 4.3 are the pillars in the imaginary columns). Let Γ_0^g be the boundary on the exterior wall corresponding to the g^{th} gap between the imaginary pillars. Since the velocity in a gap is parabolic, we impose the parabolic velocity profile on Γ_0^g and zero velocity on the rest of the exterior wall and on the interior pillars as a Dirichlet boundary condition

$$\mathbf{U}(\mathbf{x}) = \begin{cases} U_{\max}\left(1 - \frac{y^2}{G^2}\right), & \mathbf{x} \in \Gamma_0^g \\ 0, & \mathbf{x} \in \left(\Gamma_0 \setminus \bigcup_g \Gamma_0^g\right) \bigcup \left(\bigcup_i \Gamma_i\right), \end{cases} \tag{4.1}$$

where $U_{\max}$ sets the velocity scale (see Fig. 4.3(a) for the boundary conditions $\mathbf{U}$). We demonstrate our model for a DLD device of the row-shift fraction $\epsilon = 0.1667$ in Fig. 4.8. The velocity magnitude far from the top and bottom walls seems homogeneous. Therefore, we expect to capture the behavior of the particles flowing away from the top and bottom walls accurately in one period using our model.

4.2.2 Verification

We performed a convergence study by considering one zig-zagging and one displacing RBCs: the zig-zagging RBC has a high viscosity contrast and the displacing RBC have a low viscosity contrast for the row-shift fraction $\epsilon = 0.1667$. We refined the spatial and temporal resolutions until the trajectories converged. We used the converged discretization for the pillars and the converged temporal resolution in all our simulations. As the row-shift fraction changes, the number of columns and hence the size of the exterior wall change. In order to maintain the same grid quality in all our simulations, we adjusted the discretization of the exterior wall depending on ϵ.



(a) Inclination angle for $\nu = 1$

(b) Inclination angle for $\nu = 10$

(c) Trajectory for $\nu = 1$

(d) Trajectory for $\nu = 10$

Figure 4.9: *Convergence in trajectories and inclination angles of RBCs with different viscosity contrasts. We consider a DLD device with a row-shift fraction of $\epsilon = 0.1667$ and RBCs with the capillary number $C_a = 0.648$ and two different viscosity contrasts, $\nu = 1$ and $\nu = 10$. We used three different resolutions Table 4.2. (a,b) Inclination angles of RBCs during their motion; (c,d) positions of the RBCs centroids. x and y positions are normalized by the length of a period of the device.*

The device we considered for the convergence study consists of 8 rows and $\lceil 1.5(1/0.1667) \rceil = 9$ columns of pillars. Since RBC shows two different transport modes in a DLD device depending on its viscosity contrast and capillary number, we

Table 4.2: *List of spatio-temporal resolutions used in the convergence study. N_Γ is the number of points on the exterior wall, N_γ is the number of points per pillar, N is the number of points on RBC, ρ_{AL} is the error tolerance for the adaptive time stepping and sets the temporal resolution. The results of the convergence study are on Fig. 4.9.*

Resolution #	N_Γ	N_γ	N	ρ_{AL}
1	2784	48	48	5E-4
2	3712	64	64	1E-4
3	5568	96	96	5E-5

wanted to run our numerical tests for both transport modes. That is why we simulated the flow of two RBCs with viscosity contrasts of $\nu = 1$ and $\nu = 10$ with the same capillary number $C_a = 0.648$. It turns out that under these conditions the RBC with $\nu = 1$ displaces and the one with $\nu = 10$ zig- zags. In our numerical scheme, the spatial resolution is set by the number of points discretizing the boundary of the exterior wall, N_Γ, the number of points discretizing the boundary of each pillar N_γ and the number of points discretizing the RBC's membrane N. The temporal resolution is determined by the error tolerance ρ_{AL}, i.e. the smaller the tolerance is, the smaller the time step sizes are (so the finer the temporal resolution is). We started with a certain spatial and temporal resolution, then increased both together (see Table 4.2 for the resolutions). Here, N_Γ and N_γ are chosen such that the maximum arc-length spacing is the same for the exterior wall and the pillars. We present the inclination angles and trajectories of the RBCs in Fig. 4.9. As expected, the RBC with $\nu = 1$ displaces (see Fig. 4.9(c)) and the one with $\nu = 10$ zig-zags (see Fig. 4.9(d)). This is captured with all the resolutions in Table 4.2. However, the resolution #1 leads the displacing RBC to flip at $x/L \approx 0.38$. The zig-zagging RBC also flips at $x/L \approx 0.78$ with this resolution. At the higher resolutions the RBCs do not flip at these locations. Additionally, both the

trajectories and the inclination angles given by the last two resolutions in Table 4.2 agree well. Therefore, we chose the resolution # 2, i.e. $N_\Gamma = 3712$, $N_\gamma = 64$, $N = 64$ and ρ_{AL} =1E-4. As we changed the row-shift fraction, we scaled the number of points for the exterior wall such that the maximum arc-length spacings of a pillar and the exterior wall were the same.

4.2.3 Validation

The separation of rigid spherical particles using a DLD device of circular pillars is well understood [71, 72, 179]. An empirical formula for the critical particle size is given as a function of the gap size G and row-shift fraction ϵ [33]

$$D_c = 1.4 G \epsilon^{0.48} \tag{4.2}$$

where D_c is the critical diameter. Particles with diameters $D > D_c$ displace while those with $D < D_c$ zig-zag. In order to validate our DLD model we simulated the separation of rigid spherical particles. We considered four different DLD devices with the same gap $G = 10\mu$m and different row-shift fractions $\epsilon \in \{0.05, 0.1, 0.167, 0.25\}$. We had 8 rigid circular particles with $D = \{2, 3, 4, 5, 6, 7, 8, 8.5\}\mu$m flowing through these DLD devices. We observed the trajectories of the particles and determined whether they zig-zag or displace in one period. We present the results of the validation study in Fig. 4.10. We also plot the line for the ratio of the critical diameter to the gap D_c/G given by the empirical formula (4.2). As expected, the particles with diameters greater than D_c (above the line in Fig. 4.10) displace while those with smaller diameters (below the line) zig-zag. Our numerical results agree well with the experimental results [33, 72].

This validates our 2D model and proves that it can capture the underlying physics of the particle separation in DLD devices.

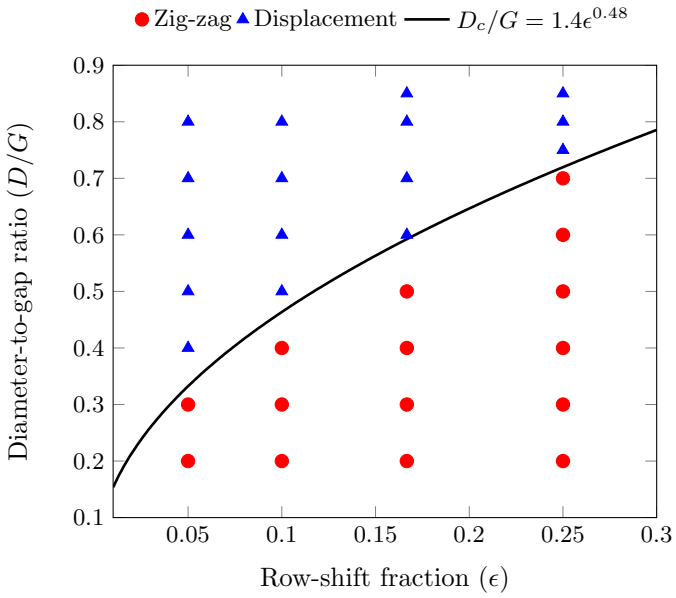


Figure 4.10: *Phase diagram for rigid spherical particles in DLD devices of circular pillars as a function of the row-shift fraction ϵ and the particle diameter-to-gap ratio D/G. We also plot the critical particle size given by the empirical formula (4.2) (solid black line).*

4.2.4 Dimensionless Numbers

Here, we summarize the parameter values that we will use in our experiments. The parameter selection is related to separating normal and abnormal RBCs based on differences in their deformability. The deformability of a cell depends on the cell's membrane elasticity, cytoplasmic viscosity and the imposed flow. These can be combined to a single non-dimensional number, the capillary number:

$$C_a = \frac{\eta_{\text{out}} R_{\text{eff}}^3}{\kappa_b} \frac{U_{\max}}{G/2} \tag{4.3}$$

where η_{out} is the viscosity of the exterior fluid, R_{eff} is the effective radius of the cell ($R_{\text{eff}} = \sqrt{A/\pi}$ in which A is the area enclosed by the cell), $U_{\max}$ is the maximum velocity of the unperturbed Poiseuille flow in a gap (4.1), κ_b is the bending stiffness and G is the gap width. The higher the capillary number is, the more deformable the cell is. Typically, the bending stiffness and cell thickness (corresponding to its effective size in deep devices) for a healthy red blood cell are $\kappa_b = 10^{-19}$Nm [159, 160] and $R_{\text{eff}} = 2.5\mu$m [133], respectively. The viscosity of the blood plasma is $\eta_{\text{out}} = 1.2$mPas. As far as the reported experiments [115], the order of $U_{\max}$ ranges between 10μm/s and 10mm/s in the DLD devices used to separate the blood components. In those devices the gap width is in the order of $\mathcal{O}(1\mu\text{m})$, which is also the size in our study. Based on those typical values, the capillary number for a healthy RBC is $C_a \in [0.0375, 375]$. For the diseased RBC the capillary number decreases to its 1/10 since the stiffness increases ten-fold [159]. Therefore, we consider the capillary numbers in the range of $[0.0038, 37.5]$. We set the capillary number by adjusting the bending stiffness κ_b only and fix all other parameters.

The dimensionless number quantifying the cytoplasmic viscosity of an RBC η_{in} is the viscosity contrast

$$\nu = \frac{\eta_{\text{in}}}{\eta_{\text{out}}}. \tag{4.4}$$

An RBC's cytoplasmic viscosity is a nonlinear function of the mean corpuscular hemoglobin concentration (MCHC), which is the concentration of the hemoglobin per unit volume of an RBC [7]. The viscosity varies from one cell to the other even within the same organism due to age because the MCHC increases as the RBC gets older. For a young red blood cell MCHC is around 32g/dl, which results in $\eta_{\text{in}} = 5 - 7$mPas, i.e., the viscosity

contrast is $\nu = 4 - 6$. If the MCHC increases to 40g/dl, the viscosity increases almost four-fold [26, 120]. Viscosity contrast is inversely proportional to cell's deformability. So, the smaller the viscosity contrast is, the more deformable the cell is. We consider $\nu \in [0.1, 100]$ in our study and set the viscosity contrast by changing the interior viscosity η_{in}.

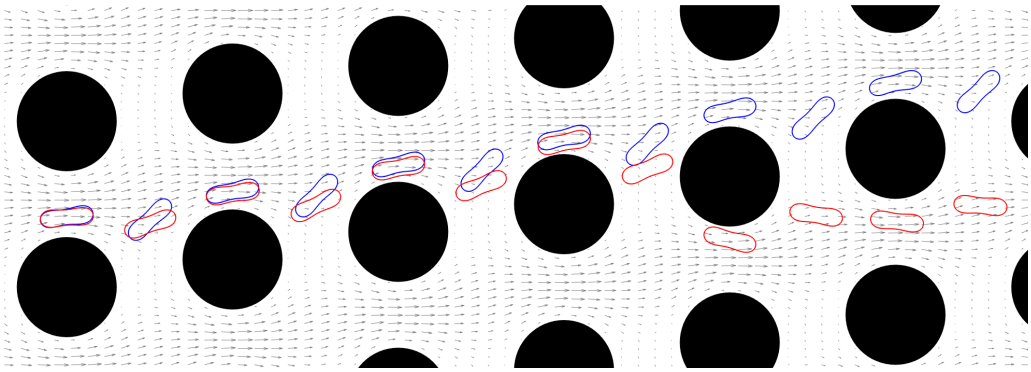
Following [168] we define a time scale as the deformation time scale driven by the imposed flow, which is the inverse of the shear gradient in a gap

$$\tau = \frac{G/2}{U_{\text{max}}}. \tag{4.5}$$

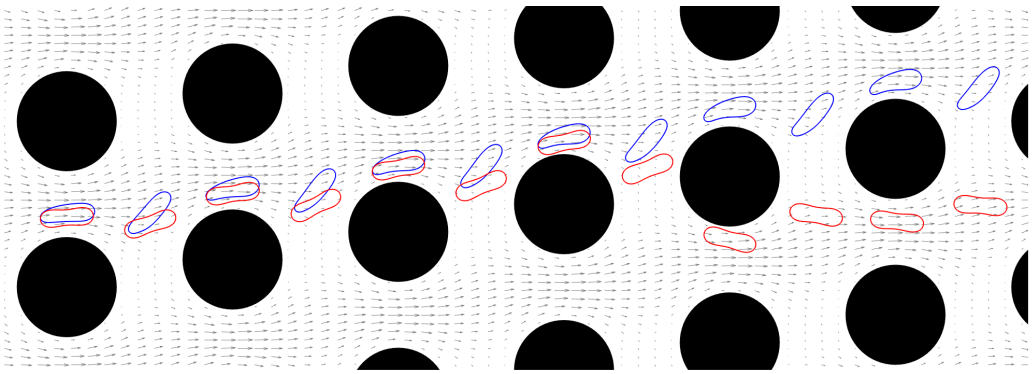
The third dimensionless number is the row-shift fraction ϵ of the DLD device. We adjust ϵ by changing the row-shift $\Delta\lambda$ and the other DLD parameters remain the same.

4.3 Results

First, we investigate the effects of the capillary number in Section 4.3.1 and the viscosity contrast in Section 4.3.2 on the dynamics of red blood cells in deep devices. Then in Section 4.3.3, we present phase diagrams for the transport modes as a function of the capillary number C_a, the viscosity contrast ν and the row-shift fraction ϵ. Finally, we study the cell separation in the flow of dense RBC suspensions through the DLD devices in Section 4.3.4.



(a) $C_a = 0.034$ (red cell) and $C_a = 0.34$ (blue cell)



(b) $C_a = 0.034$ (red cell) and $C_a = 34$ (blue cell)

Figure 4.11: *Superimposed snapshots from the simulations of RBCs having different capillary numbers C_a and the unperturbed velocity field (shown with gray arrows). We initialize the cells at the same position and simulate them separately. We, then, superimpose the snapshots taken when the cells are in the gaps and between two consecutive gaps. The viscosity contrast and the row-shift fraction in these simulations are the same, $\nu = 1$ and $\epsilon = 0.1667$, respectively. In (a), we compare the dynamics of two stiff cells with $C_a = 0.034$ (red) and $C_a = 0.34$ (blue). While the stiffer cell (red) zig-zags, the softer one (blue) displaces. In (b), we remain the stiffest cell ($C_a = 0.034$, red one) but replace the softer one with even a softer cell ($C_a = 34$, blue one). Again, the softer cell (blue) displaces.*

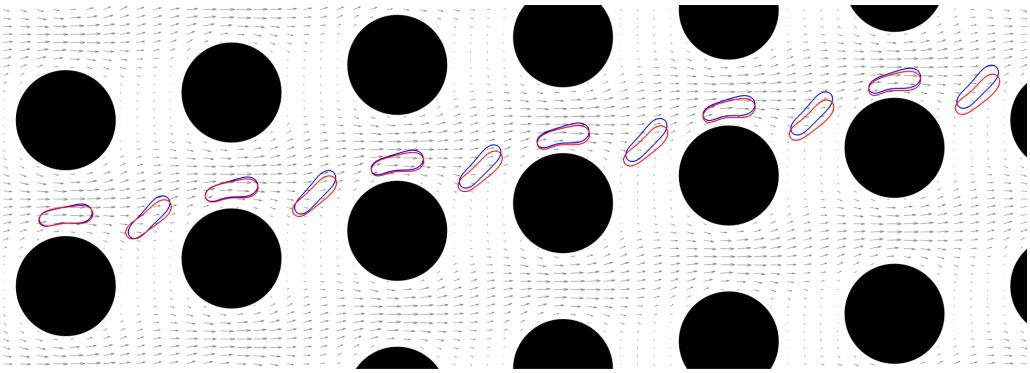
4.3.1 Effects of Capillary Number

We performed simulations of a single RBC flowing through DLD with the capillary numbers $C_a = (0.034, 0.34, 34)$. The viscosity contrast for these cells is $\nu = 1$. We considered only one device with row-shift fraction $\epsilon = 0.1667$. We present the superimposed snapshots from these simulations in Fig. 4.11 together with the unperturbed velocity field (shown with gray arrows). The snapshots are taken when the cells are in the gaps and between the consecutive gaps. The cells are initialized at the same location and have effective radius $R_{\text{eff}} = 2.4\mu$m.

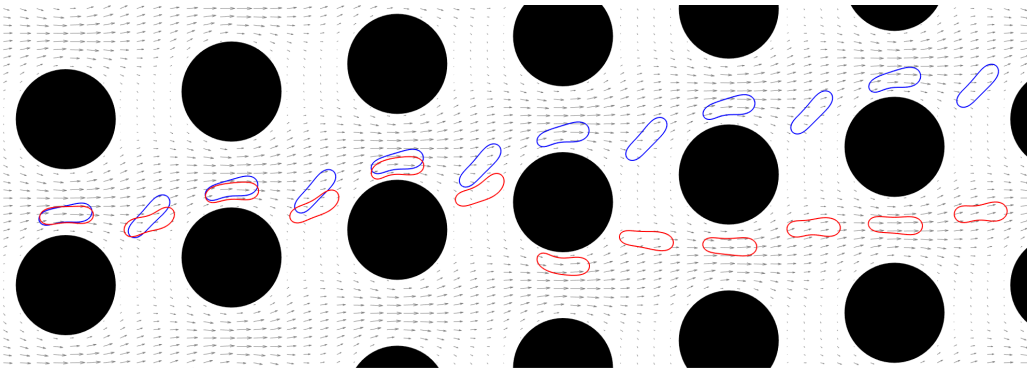
In Fig. 4.11(a) we compare two stiff cells, the red one ($C_a = 0.034$) is stiffer than the blue one ($C_a = 0.34$). While the stiffer one zig-zags, the softer one displaces. Then, in Fig. 4.11(b) we replace the softer one with an even softer cell having $C_a = 34$. The softer cell again displaces. We no discuss our observations from these simulations. In the gaps, the stiff cell (low C_a) does not deform as much as the soft one does (high C_a). The stiff cell behaves like a rigid particle and moves closer to the pillars than the softer cell does. A rigid spherical particle with a diameter less than 6μm zig-zags in a device with $\epsilon = 0.1667$ as shown in Section 4.2.3. Since the cell's effective size is less than this critical size, it zig-zags. There are several other differences in the cells' behaviors depending on their stiffness. The softer cells have an asymmetry in their shape while they are passing through a gap. They have a thick head and a thin tail in the gaps (this is more apparent for $C_a = 34$) while the stiff cell maintains its symmetric relaxed shape. The asymmetry in the soft cells' shapes and the shear gradient in the gap lead the soft cells to have a higher inclination angle with respect to the main flow direction than the stiff ones while they are moving towards the region between two

consecutive gaps.

The positive inclination angle with respect to the flow direction and the shear gradient together result in a phenomenon called cell migration, which has been extensively studied in the literature [28, 31, 50, 53, 117, 125, 168]. A cell placed near a wall migrates away from the wall in confined Poiseuille flow or shear flow due to its positive inclination with respect to the wall and the shear gradient or the curved flow lines. Here, the flow in DLD resembles the Poiseuille flow (confined flow in the gaps and almost free flow between the gaps). Therefore, a cell with positive inclination moves away from the pillars and if it is sufficiently far from the pillars (i.e., it is out of the adjacent stream swapping a lane), it displaces. As the inclination increases, it moves even farther away from the pillars. Fig. 4.11 shows that as the cell becomes softer, its inclination angle increases. That is, the cells with $C_a = (0.34, 34)$ have higher inclination angles right after the first gap than the one with $C_a = 0.034$. The softer cells maintain this positive inclination and stay away from the pillars. Additionally, the cell with $C_a = 34$ has a higher inclination angle than the one with $C_a = 0.34$ and moves farther away from the pillars than the one with $C_a = 0.34$ (see the gaps in both figures). Zhang et al. [179] also observed that RBCs stay away from pillars with square cross-sections and attributed this behavior to the fact that the flow resembles a confined Poiseuille flow and hence the cells migrate. We discuss the similarities between the flow in DLD and confined Poiseuille flow in Section 4.4 and compare the cells' inclination angles for various capillary numbers and viscosity contrasts.



(a) $\nu = 1$ (blue cell) and $\nu = 2$ (red cell)



(b) $\nu = 1$ (blue cell) and $\nu = 5$ (red cell)

Figure 4.12: *Superimposed snapshots from the simulations of RBCs having different viscosity contrasts ν and the unperturbed velocity field (shown with gray arrows). We initialize the cells at the same position and simulate them separately. We, then, superimpose the snapshots taken when the cells are in the gaps and between two consecutive gaps. The capillary number and the row-shift fraction in these simulations are $C_a = 3.41$ and $\epsilon = 0.1667$, respectively. In (a) we compare the dynamics of the less viscous cells with $\nu = 1$ (blue) and $\nu = 2$ (red). Both of these cells displace. In (b) we compare the cell having $\nu = 1$ with that with $\nu = 5$. The cell with $\nu = 5$ corresponds to a healthy RBC and it zig-zags.*

4.3.2 Effects of Viscosity Contrast

In the previous section, we considered cells with no viscosity contrast. We now want to investigate how the cells' dynamics in DLD depend on their viscosity

contrast. We performed simulations of a single RBC flowing through DLD with the viscosity contrast values $\nu = (1, 2, 5)$. The capillary number for these cells is $C_a = 3.41$. We considered the same device as in the previous section, i.e., the row-shift fraction is $\epsilon = 0.1667$. We present the superimposed snapshots from these simulations in Fig. 4.12 together with the unperturbed velocity field (shown with gray arrows). Here, the snapshots are taken when the cells are passing through the gaps and between two consecutive gaps. The cells are initialized at the same location and have an effective radius $R_{\text{eff}} = 2.4\mu$m.

The RBCs here have a moderate capillary number so that they can be considered more deformable than the stiff cells in the previous section. In Fig. 4.12(a) we compare cells with no viscosity contrast (i.e., $\nu = 1$, blue) and with the viscosity contrast $\nu = 2$ (red). Here, both cells displace. As observed in the previous section, both cells have thick heads and thin tails in the gaps. Additionally, they have positive inclination angle with respect to the flow direction in the gaps. Their inclination angle increases when they are between two consecutive gaps. Then, due to this asymmetry and the shear gradient, they migrate away from the pillars. Since they stay sufficiently far from the pillars, they displace. In Fig. 4.12(b) we compare a cell having no viscosity contrast with one having a viscosity contrast $\nu = 5$. The cell with $\nu = 5$ at this capillary number might be considered a healthy cell and it zig-zags in this device. Although the more viscous cell (red one) still maintains a positive inclination with respect to the flow direction, this angle is lower than the one with no viscosity contrast (see the cells right after the first gap). As a result of having smaller inclination angle, the more viscous cell cannot remain as far away from the pillars as the cell with no viscosity contrast

does (see the cells in the third gap for instance). So the more viscous cell eventually swaps a lane.

It is known that the cells show either tank-treading (moving with the same inclination) or tumbling (moving while rotating around its axis) depending on its viscosity contrast and capillary number. In free shear flows, the cells with $\nu = 5$ tumble [81]. So one might expect the same cell to tumble in DLD as well. However, the confinement reduces the inclination angle of a tank-treading cell and also delays the tumbling. It turns out that tumbling does not occur for the confinement level we have in the gaps in this study, which is $2R_{\text{eff}}/G = 0.48$ (see [84]). That is why the cells in DLD always maintain a positive inclination and the inclination decreases as the viscosity contrast increases. We discuss how the inclination angle varies with viscosity contrast in both confined Poiseuille flows and DLD in detail in Section 4.4.

Remark. Another consequence of aging besides increasing viscosity contrast is that RBCs lose surface area (arc-length in two dimensions) [120], so the reduced area increases. In shear and Poiseuille flows, the cell's migration velocity and inclination angle decrease as the reduced area increases. In the circular limit (a reduced area of 1), the cell does not deform and migrate. Since the cell dynamics in DLD is different than that in simpler flows, how the transport modes depend on the reduced area needs to be investigated. In order to shed some light on this issue we performed simulations of cells with $C_a = 3.41$ and $\nu = (1, 10)$, and DLD devices with $\epsilon = (0.1, 0.125, 0.167)$. We increased the reduced area from 0.65 to 0.75, 0.85 and 0.95. Recall that the reduced area is 0.65 for the other experiments in the study. We found that the cell with $\nu = 1$

displaces for all the ϵ values regardless of its reduced area. The cell with $\nu = 10$ zig-zags for these ϵ values when the reduced area is 0.65. For greater reduced areas the cell tumbles for $\epsilon = (0.1, 0.125)$ and displaces while it still zig-zags for $\epsilon = 0.1667$. Our conclusion is that the cell's transport mode in DLD can change with its reduced area depending on the row-shift fraction ϵ and the viscosity contrast ν. This shows that a more detailed study of the reduced area effects is required, but such a study is beyond the scope of our work.

4.3.3 Phase Diagrams

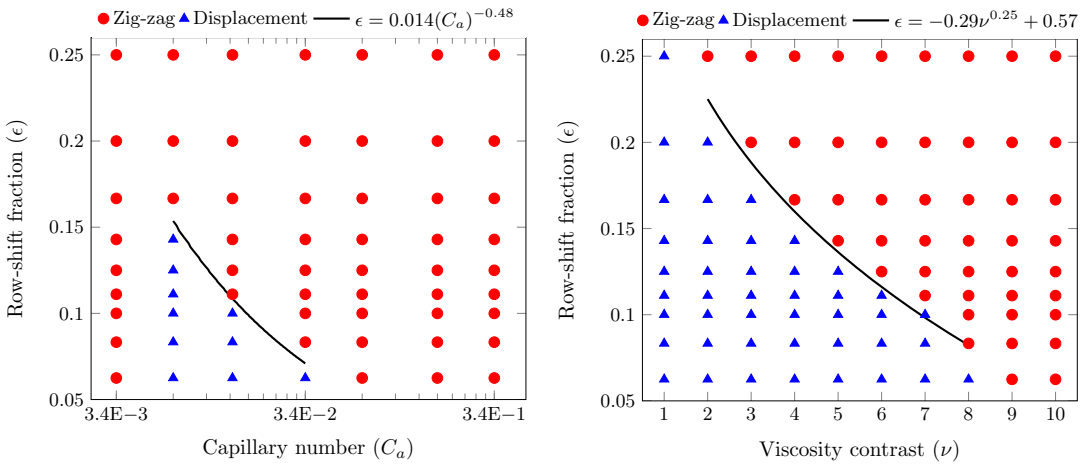
We performed an analysis in order to determine how the cells' transport modes depend on their capillary number C_a, viscosity contrast ν and the DLD's row-shift fraction ϵ. We obtained three phase diagrams for the transport modes of a single cell in DLD which are described below.

1. ϵ vs. C_a (in Fig. 4.13(a)): we fixed the viscosity contrast to $\nu = 10$ and performed simulations for $\epsilon \in [0.0625, 2.5]$ and $C_a \in [3.4 \times 10^{-3}, 3.4 \times 10^{-1}]$.

2. ϵ vs. ν (in Fig. 4.13(b)): we fixed the capillary number to $C_a = 3.4 \times 10^{-1}$ and performed simulations for $\epsilon \in [0.0625, 2.5]$ and $\nu \in [1, 10]$.

3. ν vs. C_a (in Fig. 4.14): we fixed the row-shift fraction to $\epsilon = 0.1667$ (at the top) and $\epsilon = 0.1$ (at the bottom). Then, we performed simulations for $\nu = (1, 2, 5, 8, 10, 100)$ and $C_a = [3.4 \times 10^{-2}, 3.4 \times 10^{3}]$.

We, first, discuss the phase diagrams for C_a vs. ϵ and ν vs. ϵ. Fig. 4.13(a) shows that as the capillary number increases, the transport mode shifts from zig-zag



(a) C_a vs. ϵ for $\nu = 10$

(b) ν vs. ϵ for $C_a = 3.4$

Figure 4.13: *Phase diagrams for the transport modes as a function of the capillary number C_a and row-shift fraction ϵ with $\nu = 10$ (a) and as a function of the viscosity contrast ν and ϵ with $C_a = 0.34$ (b). Red circles and blue triangles indicate zig-zagging and displacing cells, respectively. The solid line in (a) corresponds to* (4.6a) *and the one in (b) corresponds to* (4.6b). *These equations approximate the separation between two transport modes.*

to displacement and for larger capillary numbers it again shifts to zig-zag. This is not observed in Fig. 4.13(b). For low viscosity contrasts the cell displaces and as the viscosity contrast increases, the transport mode shifts to zig-zag. Since the rigidity of a cell increases as its viscosity contrast increases or its capillary number decreases, one would expect the same dynamics as a cell becomes rigid. Our results agree with this expectation. The phase diagrams show that a cell has the same transport mode in the limit of low capillary number and high viscosity contrast. This transport mode is zig-zag. This is reasonable because a very rigid cell behaves like a rigid spherical particle with a diameter equal to the cell's thickness $\approx 2.5\mu$m (i.e., it cannot migrate and its effective size is $\approx 2.5\mu$m) and the critical particle size given by (4.2) for the row-shift

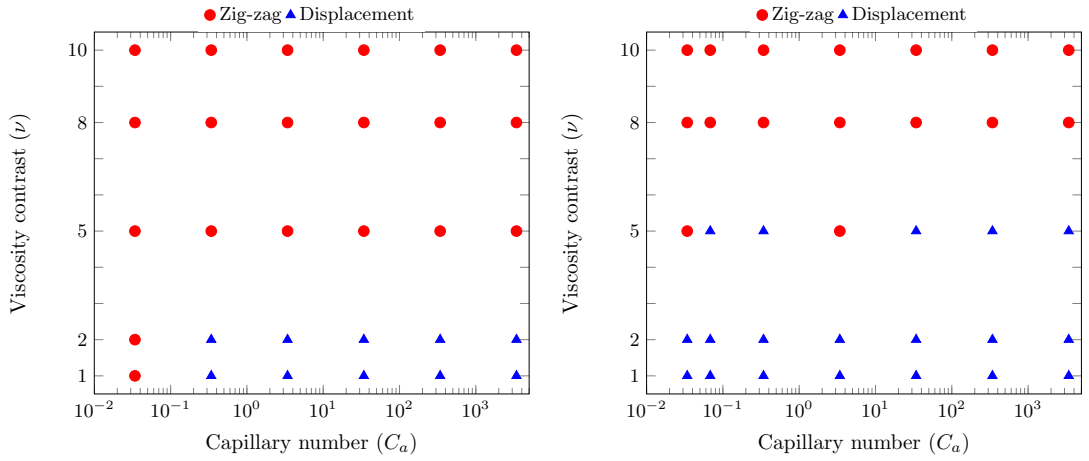
fractions considered here is greater than the cell's thickness (i.e., $D_c = 3.7\mu$m). So the rigid cell zig-zags. The reason why we observe different transport modes depending on the capillary number and viscosity contrast is that a cell's inclination angle depends on these parameters and it migrates some amount which depends on its inclination angle as observed in the previous sections. However, the rigid cell does not migrate. The curves separating the transport modes in these figures are given by a power law (4.6a) (for Fig. 4.13(a)) and (4.6b) (for Fig. 4.13(b)).

$$\epsilon = 0.014{C_a}^{-0.48}, \tag{4.6a}$$

$$\epsilon = -0.29\nu^{0.25} + 0.26. \tag{4.6b}$$

R^2 is the coefficient of determination, which indicates the goodness of a fit. The curve fits (4.6a) and (4.6b) have $R^2 = 0.92$ and $R^2 = 0.91$, respectively.

Fig. 4.14 indicates how the transport mode depends on the capillary number and viscosity contrast for a fixed row-shift fraction. Here, Fig. 4.14(a) is for the row-shift fraction $\epsilon = 0.1667$ and Fig. 4.14(b) is for $\epsilon = 0.1$. We observe that in DLD with higher row-shift fraction cells zig-zag more. That is, under the same conditions (ν and C_a) a cell zig-zags for high ϵ and displaces for low ϵ. For example, the mode is zig-zag in $\epsilon = 0.1667$ and displacement in $\epsilon = 0.1$ for $\nu = 5$ and $C_a = 3.4 \times 10^{-1}$. The reason is as the row-shift fraction increases, the width of the adjacent stream (and hence the critical size) increases. In order for a cell to displace for that case higher migration velocity (i.e. higher inclination angle) is required. That is why for the same viscosity contrasts and capillary numbers, Fig. 4.14(a) ($\epsilon = 0.1667$) has more zig-zagging cases than Fig. 4.14(b) ($\epsilon = 0.1$). For $\epsilon = 0.1667$, only the cells with viscosity contrasts



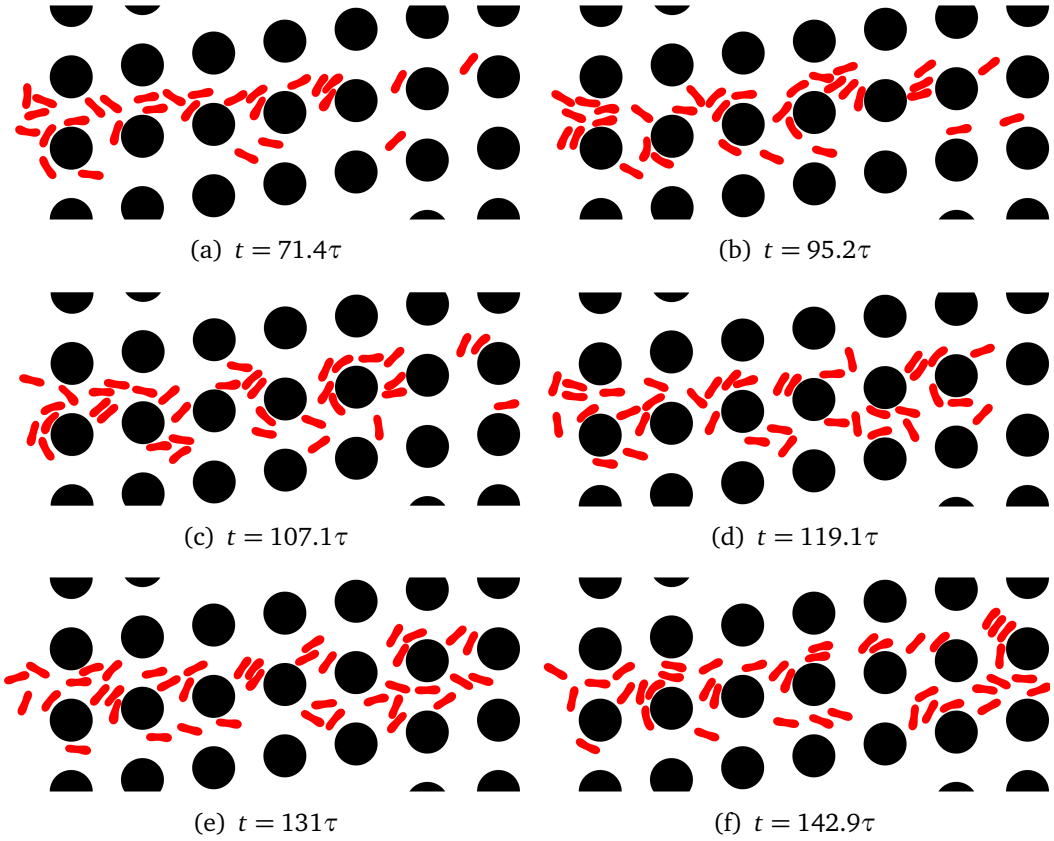
(a) C_a vs. ν for $\epsilon = 0.1667$

(b) C_a vs. ν for $\epsilon = 0.1$

Figure 4.14: *Phase diagrams for the transport modes as a function of the viscosity contrast ν and capillary number C_a for the row-shift fractions $\epsilon = 0.1667$ (a) and $\epsilon = 0.1$ (b). Red circles and blue triangles indicate zig-zagging and displacing cells, respectively.*

$\nu = (1, 2)$ displace if their capillary number is greater than 3.4×10^{-2}. So the viscosity contrast based separation is possible for these cells. However, the capillary number based separation is only possible for viscosity contrasts $\nu = (1, 2)$ and for very low capillary numbers. For $\epsilon = 0.1$, the transport mode shifts from zig-zag to displacement and then to zig-zag again as the capillary number increases for the viscosity contrast $\nu \geq 5$. So, the separation based on the capillary number is possible for higher capillary numbers than it is for $\epsilon = 0.1667$.

4.3.4 Breakdown of DLD Efficiency in Dense Suspensions



(a) $t = 71.4\tau$ (b) $t = 95.2\tau$

(c) $t = 107.1\tau$ (d) $t = 119.1\tau$

(e) $t = 131\tau$ (f) $t = 142.9\tau$

Figure 4.15: *Snapshots from a simulation of a dense RBC suspension for the row-shift fraction* $\epsilon = 0.1667$*. The suspension has an area fraction 15%. All RBCs in the suspension have the same capillary number* $C_a = 0.34$ *and viscosity contrast* $\nu = 1$*. The phase diagram in Fig. 4.13 shows that a single RBC with these parameters displaces. In its dense suspension, however, only 30% of the RBCs displace.* τ *is the time scale defined as the inverse of the shear gradient in a gap, i.e.,* $\tau = G/2U_{\max}$*. See Fig. 4.16 for the statistics of the dense suspensions.*

The DLD theory is developed for dilute suspensions assuming that the unperturbed velocity field (in the absence of particles) is not distorted when the particles are present. So that the particle-pillar interactions result in deterministic particle trajecto-

ries and hence separation. Additionally, the phase diagrams for deformability-based RBC separation in our study and the literature [97, 145] are obtained simulating a single particle in a DLD device. As the suspensions become denser, the particle-particle interactions dominate the particle-pillar interactions and separation may not occur anymore because particles with the same properties are not in the same transport modes [166]. The DLD efficiency for dense suspensions is of interest because the use of dense suspensions would reduce the operation time and remove the pre-treatment (such as dilution) requirement. Inspired by [166] we are interested in failure of the transport modes in dense suspensions of RBCs in deep devices. In order to explain the failure, let us consider a single RBC with a certain viscosity contrast and capillary number and suppose that this RBC displaces in its dilute suspension for a certain row-shift fraction. When a dense suspension of this RBC flows in the same device, not all of the cells displace. This is called failure in the displacement mode. Similarly, we define failure in the zig-zag mode, too. [166] performed 3D simulations of the flow of dense RBC suspensions in shallow devices for various capillary numbers, row-shift fractions and area fractions under a constant viscosity contrast of $\nu = 5$. The key findings of their study are:

1. As the volume fraction increases for the same row-shift fraction and capillary number, more RBCs zig-zag independent of the transport mode of a single RBC under the same conditions. So the displacement mode is more prone to failure than the zig-zag mode. Consequently, it is easier to separate large particles such as white blood cells from dense suspensions of RBCs than the small particles (e.g., platelets), since the small particles also zig-zag like the RBCs.

2. At the same volume fraction and capillary number, increasing the row-shift fraction reduces the failure rate of the displacement mode in one period of the device. However, a DLD device with small row-shift fraction must be long enough to have significant lateral separation between displacing and zig-zagging particles at the end. This requires using many periods of the device, which increases the failure rate in the overall device. Therefore, the use of small row-shift fractions does not prevent the breakdown.

Inglis et al. [73] also made the same observations in their experimental study.

Here, we performed numerical simulations of dense suspensions in deep devices. We, first, considered RBCs with two different capillary numbers $C_a = (0.0034, 0.34)$ but a fixed viscosity contrast of $\nu = 10$ for the row-shift fractions $\epsilon = (0.0833, 0.125, 0.1667)$. Secondly, we conducted the same experiments with two different viscosity contrasts $\nu = (1, 10)$ but a fixed capillary number of $C_a = 0.34$. As in the previous sections, we observed the transport mode in one period of the device. We filled the middle lane of the device with randomly initialized RBCs. Then, as the simulation went on, we initialized new RBCs at a random lateral position at the left-most point of the device randomly in time. We ran the simulations until 100 RBCs in total traveled to the end of the device. We define and calculate the area fraction as follows. Considering a circle centered at each cell's center with diameter equal to the cell's arc-length, we compute the so-called local area fraction which is the ratio of the area occupied by the cells in the circle to the area of the circle. Finally, taking the average of the local area fraction over time and cells delivers the area fraction of the simulation. In Fig. 4.15 we present the snapshots from the simulation of RBCs with $C_a = 0.34$ and $\nu = 1$ for the row-shift

fraction $\epsilon = 0.1667$. We report the numbers of zig-zagging and displacing RBCs in the dense suspensions in Fig. 4.16. The dense suspensions in our simulations had area fractions of 15%-20%. We found that one RBC interacts with approximately two to three other RBCs on average throughout its motion in our simulations. The statistics in Fig. 4.16 agree well with the findings of [73, 166]. We summarize our results:

1. The displacement mode due to either viscosity contrast or capillary number is more susceptible to failure. That is, more than 50% of the RBCs in a dense suspension zig-zag for the two largest row-shift fractions (the first two columns in Fig. 4.16) while the transport mode of an RBC with the same properties is displacement in its dilute suspension (see the first and the last rows). Whereas only 15-20% of the zig-zagging RBCs fail to zig-zag (see the second row).

2. For the smaller row-shift fractions (from left to right in the same figure), the breakdown is less pronounced. Only 25% of the displacing RBCs zig-zag in a dense suspension for the smallest row-shift fraction (see the last column).

4.4 Discussion

In this section, we investigate the underlying mechanism for the RBC separation depending on the capillary number and viscosity contrast in deep DLD devices. Since the flow in DLD resembles free shear flow (between the vertical gaps) and confined Poiseuille flow (in the vertical gaps), we compare the cell dynamics in DLD with that in these simpler flows. We present the similarities and seek to understand if these simpler

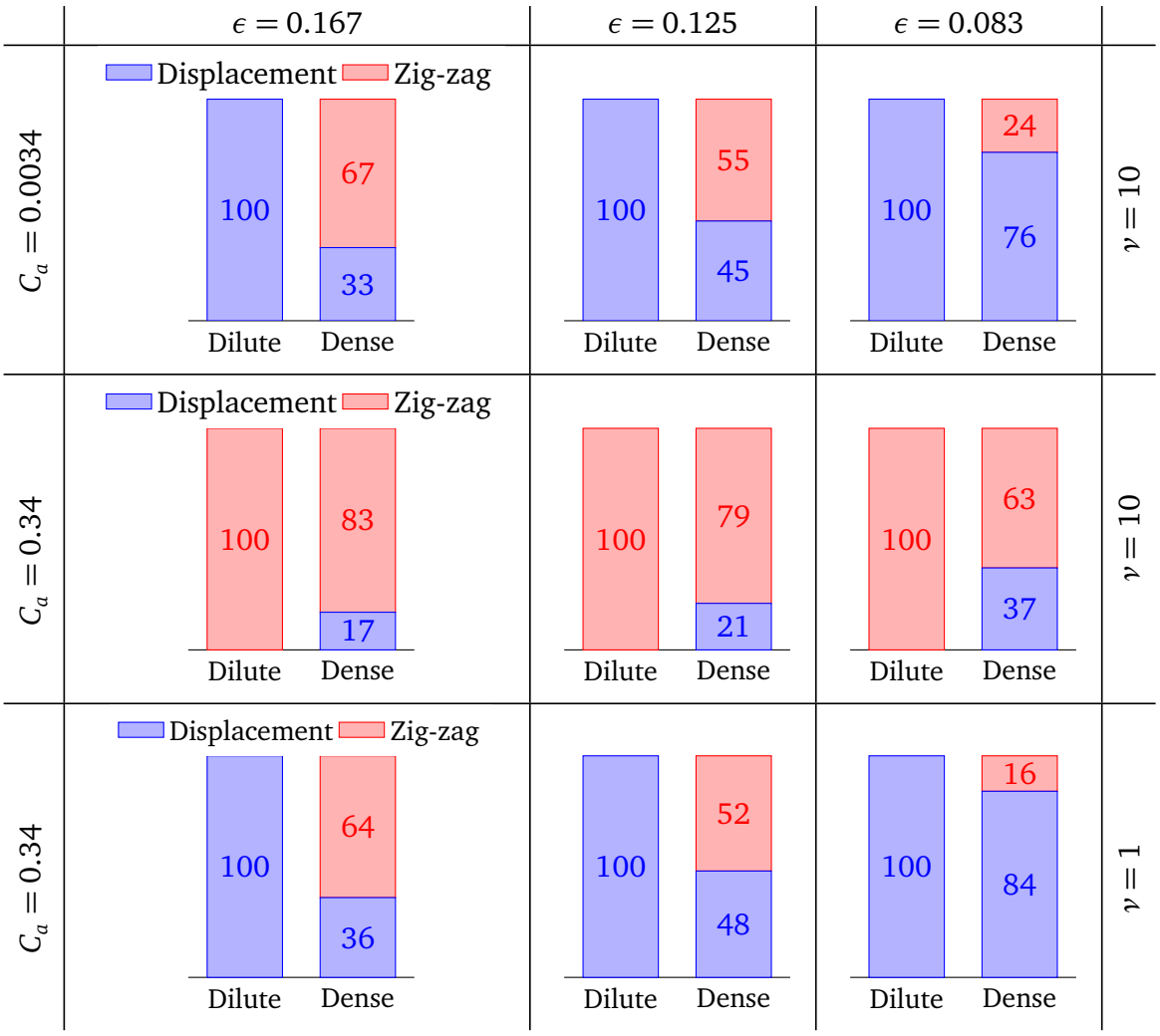


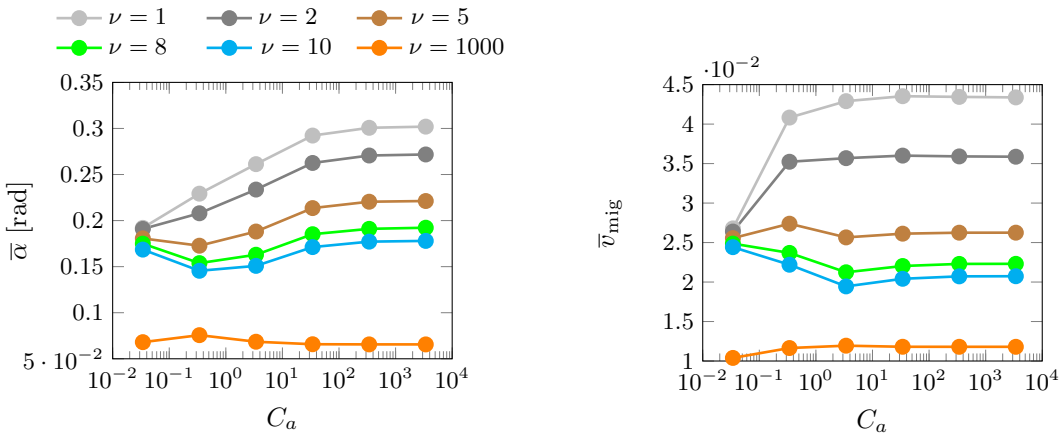
Figure 4.16: *Breakdown of the transport modes in dense RBC suspensions. We performed simulations of a single RBC (dilute suspension) and multiple RBCs (dense suspension) for the row-shift fractions* $\epsilon = (0.083, 0.125, 0.1667)$*, the capillary numbers* $C_a = (0.0034, 0.34)$ *and viscosity contrasts* $\nu = (1, 10)$*. The dense suspensions have area fractions 15 to 20%. The histogram plots show the numbers of zig-zagging and displacing RBCs (out of 100). In the dilute suspensions, a single RBC either zig-zags or displaces. Whereas in dense suspensions the same RBCs are not in the same transport modes. Results at each column are for different row-shift fractions and those at each row have different combinations of* C_a *and* ν*.*

set-ups can be used to predict whether a cell displaces or zig-zags in a particular DLD device.

4.4.1 Free Shear Flow

The results in Section 4.3 show that cells having high positive inclination angles with respect to the flow direction displace and those having lower inclination angles zig-zag. This is a result of a positive inclination angle and shear gradient which lead a cell to migrate away from the pillars. A cell's inclination angle depends on its capillary number and viscosity contrast. In free shear flow, cells tilt to a certain angle and move with that angular orientation for low capillary numbers and viscosity contrasts. This motion is called tank-treading. For high capillary numbers and viscosity contrasts, cells do not have such an angular orientation and tumble [81, 118]. In free shear flow, tank-treading cells migrate from high shear rate regions to low shear rate regions while tumbling cells do not migrate [117, 125, 168]. Henry et al. [65] investigated the sorting of cells depending on this cell dynamics and concluded that tank-treading at the viscosity contrast $\nu = 1$ results in displacement and tumbling usually leads to zig-zag. They observed tumbling between the gaps right before a cell zig-zags which we also observed for some cases. It must be noted that tumbling does not occur in the gaps in DLD due to the confinement [84].

We investigate whether the conditions (the capillary number and the viscosity contrast) causing cells to tank-tread in free shear flow lead cells to displace in DLD for any row-shift fraction. In other words, we investigate whether the conditions causing cells to tumble in free shear flow lead cells to zig-zag in DLD for any row-shift fraction.



(a) Average inclination angles

(b) Average migration velocity

Figure 4.17: *The time-averaged inclination angles $\overline{\alpha}$ and the migration velocities $\overline{v}_{\text{mig}}$ for a red blood cell in confined Poiseuille flow for various capillary numbers C_a and viscosity contrasts ν. We considered a channel having the same confinement as the gaps in our DLD simulations. We initialized a cell below the centerline and let the cell to migrate towards the center.*

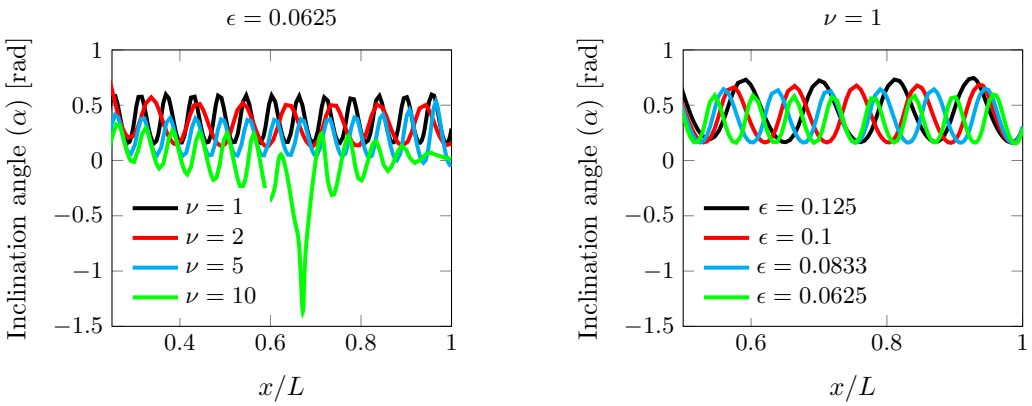
If this is true, then we can estimate the cell dynamics in DLD by simulating cells in a simpler free shear flow. However, as we can see in Fig. 4.14, a cell tank-treading in a free shear flow does not necessarily displace in DLD flow for the same capillary number and the viscosity contrast. For example, a cell that tumbles for $\nu = 5$ and $C_a = \mathcal{O}(10)$ in free shear flow displaces in the DLD for $\epsilon = 0.1$ but zig-zags for $\epsilon = 0.1667$. The main reason for that the dynamics in free shear flow cannot predict the transport mode in DLD is that the row-shift fraction information does not enter free shear flow. Additionally, the confinement in the gaps in DLD also affects this dynamics [84].

4.4.2 Confined Poiseuille Flow

Confining a cell by walls delays the cell's transition from tank-treading to tumbling and can even avoid tumbling [84]. In order to incorporate the confinement effects into a simpler model, we now consider confined Poiseuille flow, which the flow in a DLD gap resembles, and we investigate the cell dynamics. We imposed a parabolic velocity similar to the flow in the DLD gaps on the side walls of a channel confined by two parallel walls. The width of this channel is the same as the width of the gap in our DLD simulations. Thus, the confinement is $\chi = 2R_{\text{eff}}/G$. The channel is centered at the origin and the cell is initialized at $y_i = -3G/10$. We performed simulations of a cell for the same capillary numbers and viscosity contrasts that we considered in our DLD simulations, i.e., $C_a \in [3.41 \times 10^{-2}, 3.41 \times 10^{3}]$ and $\nu \in [1, 10]$. We let the cell migrate to $y_f = -G/10$. We do not allow the cell to reach the center because there it shows complex shape changes and equilibrium dynamics for this confinement [7]. Additionally, the cell in DLD usually stays near the pillars not at the center of the gaps. We measured the cell's migration velocity v_{mig} (i.e., velocity in the direction perpendicular to the flow direction) and inclination angle α with respect to the walls at various lateral positions. We present the time-averaged inclination angles $\overline{\alpha}$ and migration velocities $\overline{v}_{\text{mig}}$ for various capillary numbers C_a and viscosity contrasts ν in Fig. 4.17. Fig. 4.17(a) shows that the inclination angle varies less with the viscosity contrast for the lowest capillary number than higher capillary numbers. As the capillary number increases, the inclination angle increases. It also increases as the viscosity contrast decreases. So for lower viscosity contrasts and higher capillary numbers the inclination angle is higher. For $C_a \geq 10^2$ the inclination angle does not change significantly

with C_a. Additionally, the inclination angle increases monotonically with the capillary number for the viscosity contrasts $\nu = (1, 2)$ but it does not change monotonically with the capillary number for the viscosity contrasts $\nu \geq 5$. Fig. 4.17(b) indicates that the average migration velocity parallels the average inclination angle. A cell migrates faster for lower viscosity contrasts. The migration velocity monotonically increases with the capillary number for the viscosity contrasts $\nu = (1, 2)$ like the average inclination angle, however, this behavior vanishes for viscosity contrasts $\nu \geq 5$. Although the inclination angle still changes with C_a for $C_a = \mathcal{O}(1)$, the migration velocity does not change significantly with C_a anymore for $C_a = \mathcal{O}(1)$. The reason might be the fact that the bending relaxation time scale dominates the time scale for shear flow for $C_a > 1$. Another observation is that a very rigid cell (i.e., the viscosity contrast $\nu = 1000$) also has a positive inclination with respect to the walls but the angle is an order of magnitude smaller than that for softer cells. This results in a very small migration velocity for the stiff cell. Overall, similar to the DLD flow examples we do not observe tumbling for this confinement in confined Poiseuille flow, which agrees with [84]. Based on these results we expect that the transport mode in DLD depends on whether the degree of a cell's inclination results in enough migration to keep the cell far away from the pillars. We now discuss how the inclination angle of a cell depends on C_a, ν and ϵ in DLD flows and compare it with the above results for confined Poiseuille flows.

We demonstrate how the inclination angle of a cell changes as the cell flows through a DLD device in Fig. 4.18. In Fig. 4.18(a), we vary the viscosity contrast ν by setting the capillary number to $C_a = 0.34$ and row-shift fraction to $\epsilon = 0.0625$. In Fig. 4.18(b) we vary the row-shift fraction for a fixed capillary number $C_a = 0.34$



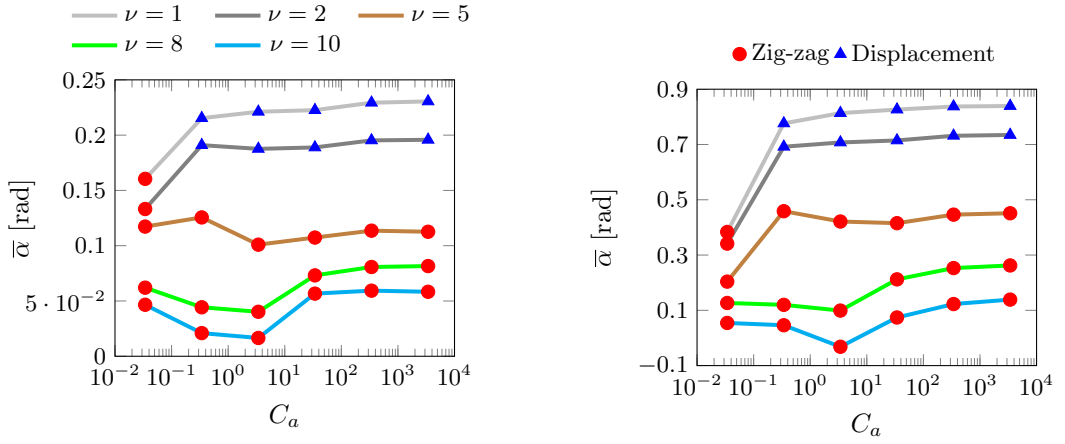
(a) Inclination angles for $\epsilon = 0.0625$

(b) Inclination angles for $\nu = 1$

Figure 4.18: *Inclination angles α of RBCs for various viscosity contrasts ν at a fixed row-shift fraction $\epsilon = 0.0625$ (a) and for various row-shift fractions at a fixed viscosity contrast $\nu = 1$ (b). The capillary number is the same, $C_a = 0.34$. The RBCs with $\nu = 1, 2, 5$ displace and the one with $\nu = 10$ zig-zags. The displacing RBCs have positive inclination angles alternating between two values (see (a)). The minimum angle is attained when the cell is in the gap and the cell reaches the maximum angle when it is between two gaps. The minimum and maximum angles reduce as the viscosity contrast increases. The maximum angle depends also on the row-shift fraction ϵ. As the row-shift fraction increases the maximum angle increases, however, the minimum angle does not change significantly (b). The x-coordinates are normalized by the lengths of the DLD devices L.*

and viscosity contrast $\nu = 1$. All these cells displace except the one with $\nu = 10$ in Fig. 4.18(a). The displacing cells attain their minimum angles in the gaps and tilt to their maximum angles between two consecutive gaps. These maximum angles are 0.58rad for $\nu = 1$, 0.5rad for $\nu = 2$ and 0.37rad for $\nu = 5$. These values are close to the steady inclination angles of the RBCs having these viscosity contrasts in an unbounded shear flow [18, 81]. However, the displacing RBCs' maximum inclination angles depend also on the row-shift fraction. Fig. 4.18(b) indicates that as the row-shift fraction increases, the maximum inclination angle increases. The reason for this

is that, as the row-shift fraction increases, the angle between the direction of the flow between the gaps and the main flow direction increases. This results in a higher maximum inclination angle. Additionally, Fig. 4.18(b) shows that the minimum angle does not change significantly with the row-shift fraction. This is because the flow in the gap is always in the main flow direction no matter what the row-shift fraction is (see Fig. 4.8). The minimum inclination angle depends only on the viscosity contrast and the capillary number.

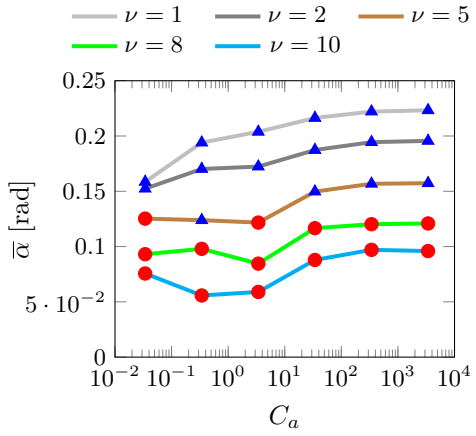


(a) Average inclination angles in gaps

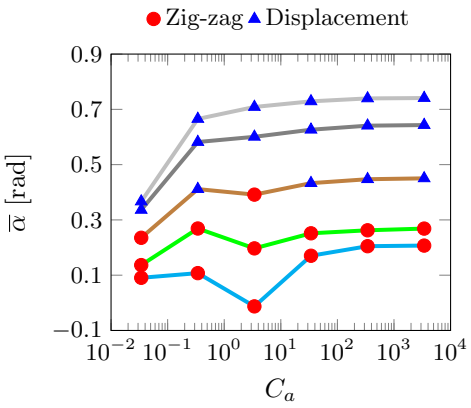
(b) Average inclination angles between gaps

Figure 4.19: *The average inclination angles $\overline{\alpha}$ of RBCs in the gaps and between the gaps for the row-shift fraction $\epsilon = 0.1667$. The results are from the simulations we performed for the phase diagram in Fig. 4.14(a). Red circles and blue triangles indicate the zig-zagging and displacing cells, respectively. The displacing cells have higher inclination angles than the zig-zagging ones. That is, the displacing cells have $\alpha > 0.16$ rad in the gaps and $\alpha > 0.6$ rad between the gaps. Whereas the zig-zagging ones have $\alpha < 0.16$ rad in the gaps and $\alpha < 0.5$ rad between the gaps.*

Since the inclination angle of a cell is much higher when the cell is between the gaps than when the cell is in the gaps, we compute the average inclination angles in the gaps and between the gaps, separately. We present these average inclination angles



(a) Average inclination angles in gaps



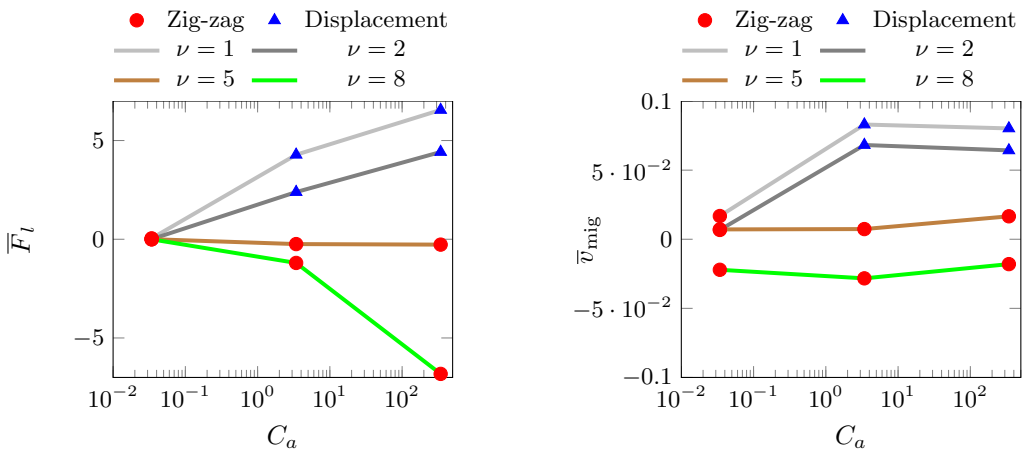
(b) Average inclination angles between gaps

Figure 4.20: *The average inclination angles $\overline{\alpha}$ of RBCs in the gaps and between the gaps for the row-shift fraction $\epsilon = 0.1$. The results are from the simulations we performed for the phase diagram at the top in Fig. 4.14(a). Red circles and blue triangles indicate the zig-zagging and displacing cells, respectively.*

$\overline{\alpha}$ as a function of the capillary number and viscosity contrast for the row-shift fractions $\epsilon = 0.1667$ in Fig. 4.19 and for $\epsilon = 0.1$ in Fig. 4.20. The figures on the left demonstrate the average angles in the gaps and those on the right show the angles between the gaps. We indicate the zig-zagging and displacing cells with red circles and blue triangles, respectively. The results for $\epsilon = 0.1667$ (see Fig. 4.19) show that the average inclination angle is higher for low viscosity contrasts and increases with the capillary number for $\nu = (1, 2)$. The average inclination angle does not monotonically change with the capillary number for $\nu \geq 5$. In this sense, the cell dynamics in DLD is similar to that in the confined Poiseuille flow (see Fig. 4.17). While the average angle is less in the gap than in the channel flow, the average angle between the gaps is much higher than the angle in the confined Poiseuille flow due to a weaker confinement between the

gaps. So, since a cell periodically moves from strong confinement in the gaps than to weak confinement between the consecutive gaps, it shows more complicated dynamics in DLD than in the confined Poiseuille flow. Additionally, since the row-shift fraction information does not appear in the confined Poiseuille flow, the transport modes of cells in DLD cannot be captured by this simpler set-up. For example, while the cells having average inclination angles in the gaps greater than 0.16rad zig-zag for $\epsilon = 0.1667$, those with the average angles less than 0.16rad displace for $\epsilon = 0.1$.

We made another observation that supports the complexity of the flow in DLD. Figure 4.19 shows that the displacing cells have higher inclination angles than the zig-zagging ones both in the gaps and between the gaps for $\epsilon = 0.1667$. Then, one might generalize and state that the displacing cells always have higher inclination angles than the zig-zagging ones. However, the results for $\epsilon = 0.1$ in Fig. 4.20 present a counter-example. That is, the higher inclination angle in the gaps does not guarantee that the cell displaces. For example, the average inclination angle in the gaps for $\nu = 5$ and $C_a = 0.034$ (the lowest capillary number) is slightly higher than the angle for the same viscosity contrast but the next capillary number $C_a = 0.34$ (see Fig. 4.20(a)). However, the cell with the lowest capillary number zig-zags and the one with a higher capillary number displaces. The reason is the cell with a higher capillary number has higher average inclination angle between the gaps than the one with the lowest capillary number (see Fig. 4.20(b)). Although neither free shear flow or confined Poiseuille flow can capture the transport mode of a cell in DLD devices, it is helpful to understand the cell dynamics in these flows to explain the underlying mechanism for the cell separation in DLD.



(a) Average pseudo-lift

(b) Average migration velocity

Figure 4.21: *The time-averaged pseudo-lift* $\overline{F}_l$ (4.7) *and migration velocities* $\overline{v}_{\text{mig}}$ *for RBCs as a function of the capillary number* C_a *and viscosity contrast* ν *for the row-shift fraction* $\epsilon = 0.1667$. *While the x-axis corresponds to* C_a, *lines with different colors correspond to* ν. *Red circles and blue triangles indicate zig-zagging and displacing cells, respectively.*

4.4.3 Pseudo-lift

In inertial flows (i.e., $Re \gg 1$) the net force in the direction perpendicular to the free-stream velocity causes a body to drift in that direction. This force is called lift F_l and given by

$$F_l(t) = \int_\gamma (\mathbf{Tn})\cdot\mathbf{k}\, d\gamma, \tag{4.7}$$

where γ is the surface of the body, $\mathbf{T}$ is total hydrodynamic stress on the body, $\mathbf{n}$ is the normal direction on the surface and $\mathbf{k}$ is a unit vector perpendicular to the free-stream velocity. In non-inertial flows (i.e., $Re \ll 1$) such as the flow of RBCs in DLD in our study, the net lift force on a body given by (4.7) turns out to be zero. By modifying the term $\mathbf{k}$ in (4.7) we define the so-called pseudo-lift for the flow of cells to quantify the cell migration in DLD. We compute the pseudo-lift $F_l(t)$ at the time t using the

same equation (4.7). Instead of a constant $\mathbf{k}$ along the boundary, we let $\mathbf{k}$ vary along the boundary such that it is defined as a unit vector perpendicular to the velocity on γ. The other terms in (4.7) remain the same. That is, $\mathbf{T}$ is the total hydrodynamic stress on the cell's boundary γ (see below for the integral equation formulation to compute $\mathbf{T}$) and $\mathbf{n}$ is the unit normal vector on γ. Since $\mathbf{k}$ varies along the boundary γ, there is a net pseudo-lift on a migrating cell. We, then, define the time-averaged pseudo-lift $\overline{F}_l$

$$\overline{F}_l = \frac{1}{T}\int_{t_i}^{t_f} F_l(t)dt, \tag{4.8}$$

where $T = t_f - t_i$. Here, we integrate between the time t_i when the cell is passing through the second gap and the time t_f which is either when a displacing RBC is at the end of a period or when a zig-zagging RBC switches a lane. Since $\mathbf{Tn}$ scales with $\kappa_b R_{\text{eff}}$ (see (2.5)), we normalize $\overline{F}_l$ with $\kappa_b R_{\text{eff}}$. In Fig. 4.21(a) we present the time-averaged pseudo-lift on the cells with the capillary numbers $C_a = (3.41 \times 10^{-2}, 3.41, 3.41 \times 10^2)$ and viscosity contrasts $\nu = (1, 2, 5, 8)$ for the row-shift fraction $\epsilon = 0.1667$. In Fig. 4.21(b) we also show the time-averaged migration velocity $\overline{v}_{\text{mig}}$ for these cells. We computed the velocity of a cell's center in the y-direction at each time step and averaged the velocity as we did it for the pseudo-lift in (4.8). The time-averaged inclination angles corresponding to these cells are already presented in Fig. 4.19. In these figures we also marked the zig-zagging and displacing cells with red circles and blue triangles, respectively. The zig-zagging cells have pseudo-lift and migration velocity either very close to zero or less than zero. Whereas, the displacing cells have an order of magnitude greater pseudo-lift and migration velocity. The average migration velocity seems to depend on the capillary number and viscosity contrast as the average inclination angle does so (See Fig. 4.19). That is, the average migration velocity

monotonically increases and reaches a plateau for $\nu = (1,2)$, and it decreases then increases for $\nu = (5,8)$. Although the pseudo-lift does not exactly follow this behavior, it can be used as a measure of the cell migration.

Computing total stress on a membrane. In order to compute the hydrodynamic lift on a cell's membrane we need to compute the total stress $\mathbf{T} = -p\mathbf{I} + \eta(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$. We revisit the integral equation formulation for the stress given in [142]. The stress $\mathbf{T}^S$ of the single-layer potential $\mathbf{u}(\mathbf{x}) = \mathscr{S}[\mathbf{f}](\mathbf{x})$ is

$$\mathbf{T}^S[\sigma](\mathbf{x}) = \frac{1}{\pi}\int_\gamma \frac{\mathbf{r}\cdot\sigma}{\rho^2}\frac{\mathbf{r}\otimes\mathbf{r}}{\rho^2}\mathbf{f}ds_\mathbf{y}, \tag{4.9}$$

which gives the stress at $\mathbf{x}$ due to a cell with an interfacial force $\mathbf{f}$. The stress $\mathbf{T}^D$ of the double-layer potential $\mathbf{u}(\mathbf{x}) = \mathscr{D}[\zeta](\mathbf{x})$ is

$$\mathbf{T}^D[\sigma](\mathbf{x}) = \frac{1}{\pi}\sum_{q=1}^{M}\int_{\gamma_q}\left(\frac{\mathbf{n}\cdot\zeta_q}{\rho^2}\sigma - \frac{8}{\rho^6}(\mathbf{r}\cdot\mathbf{n})(\mathbf{r}\cdot\zeta_q)(\mathbf{r}\cdot\sigma)\mathbf{r} + \frac{\mathbf{r}\cdot\mathbf{n}}{\rho^4}(\mathbf{r}\otimes\zeta_q + \zeta_q\otimes\mathbf{r})\sigma\right) + \left(\frac{\mathbf{r}\cdot\zeta_q}{\rho^4}(\mathbf{r}\otimes\mathbf{n} + \mathbf{n}\otimes\mathbf{r})\sigma\right)ds_\mathbf{y}, \tag{4.10}$$

which is the stress at $\mathbf{x}$ due to the pillars with density ζ. Total stress is the summation of the stresses due to a cell, pillars and an exterior wall.

Remark. The normal stress difference (i.e. $N = \langle T_{xx}\rangle - \langle T_{yy}\rangle$), where the angle bracket $\langle\cdot\rangle$ denotes volume average, has been used to quantify cell migration in shear and Poiseuille flows [49, 50]. In these flows, the normal stress difference is positive during migration of a tank-treading cell and becomes zero when migration ends. So, the positive normal stress difference indicates cell migration. A tumbling cell's inclination

angle oscillates periodically, which results in a nonlinear behavior of N in time. In DLD flows, a cell's inclination angle oscillates like a tumbling cell (see Fig. 4.18). As a result of this, the normal stress difference shows a nonlinear behavior in time in DLD flows. In order to investigate if the normal stress difference indicates whether a cell displaces or zig-zags, we computed the its time average for several cases: cells with $C_a = (0.034, 3.4, 340)$ and $\nu = (1, 2, 5, 8)$ for $\epsilon = 0.1667$. We found positive average normal stress difference for the displacing cells (those with $C_a = (3.4, 340)$ and $\nu = (1, 2)$. However, the average normal stress difference turned out to be non-negative or nearly zero always for zig-zagging cells. The reason is that, although a zig-zagging cell maintains a positive inclination angle most of the time (hence positive average normal stress difference), it may not be able to generate sufficient migration to displace. Overall, we have not observed any stronger correlation between a cell's transport mode and the average normal stress difference than the one between the mode and the pseudo-lift.

4.5 Conclusions

We have studied sorting RBCs by their deformability (i.e., the membrane stiffness and interior fluid viscosity) using deep DLD devices. We have quantified cells' deformability with the dimensionless numbers, capillary number C_a and viscosity contrast ν. We have performed a systematic study to map out the parameter space for the transport modes as a function of C_a, ν and the row-shift fraction ϵ describing the DLD geometry. We have observed that an RBC is either in the displacement mode with a steady angular orientation or in the zig-zag mode depending on its C_a and ν. This leads

RBCs to have a net non-zero or almost zero lateral displacement as they leave a DLD device and hence enables the sorting. We have discussed the underlying mechanism of the sorting. Since the RBC dynamics in DLD resembles that in confined Poiseuille flows, we have compared the cell dynamics in these flows. RBCs either tank-tread with a steady inclination angle or tumble in confined Poiseuille flows. Lateral confinement in the gaps in DLD, however, restricts tumbling and all the cells move with a positive inclination with respect to the flow direction. The degree of the inclination angle depends on the cells' capillary number and viscosity contrast. Positive inclination with respect to the flow and the shear gradient results in cell migration towards a low shear gradient region. Cells having higher inclination angles stay farther away from the pillars than those having smaller inclination angles and hence displace. Finally, we define the so-called pseudo-lift to quantify the degree of migration. We have found that strong positive pseudo-lift acts on the displacing RBCs while either weak or negative lift force acts on the zig-zagging RBCs. We have also performed simulations with dense RBC suspensions for various capillary numbers, viscosity contrasts and row-shift fractions. Our findings agree well with the numerical [166] and experimental [73] studies. Most of the cells zig-zag in dense suspensions no matter what their deformability is. Hence, the deformability-based sorting fails.

Chapter 5

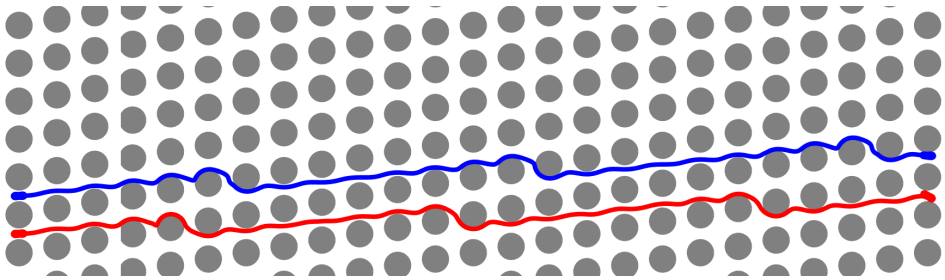
Optimal Design of Deterministic Lateral Displacement Device for Viscosity-Contrast-Based Cell Sorting

In this chapter[1] we discuss a design optimization problem for deterministic lateral displacement (DLD) device to efficiently sort same-size red blood cells (RBCs) by their deformability, in particular, their viscosity contrast between the fluid in the interior and the exterior of the cells. We explained deformability-based sorting of RBCs using DLD in Chapter 4. Here, we propose an objective function such that a design minimizing it delivers designs providing efficient sorting of cells with arbitrary viscosity contrast values. The objective function is evaluated by simulating the cell flows through a device using our 2D vesicle model described in Chapter 2. We solve the optimization problem using a stochastic optimization algorithm. Since the algorithm converges in $\mathcal{O}(1000)$ iterations and our high-fidelity DLD model (see Section 4.2.1) is expensive to evaluate the objective function, we propose a low-fidelity DLD model that enables fast solution of the problem. Finally, we present several scenarios where solving the optimization problem finds designs that can separate cells with similar viscosity contrast values. These designs have cross sections that have features similar to a triangle. To the best of our knowledge, this is the first study which poses designing a DLD device as a

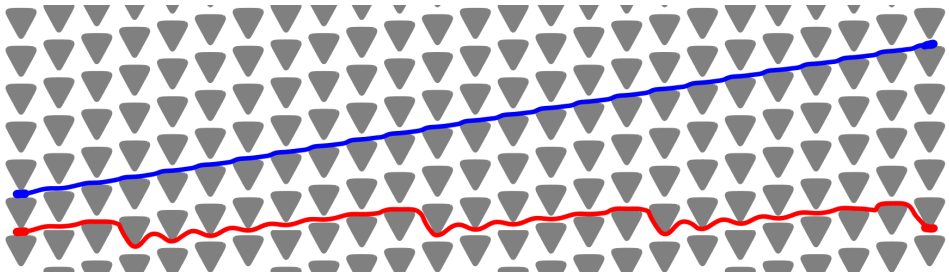
[1] This chapter is based on work that has been published in [76]. The authors equally contributed.

constrained optimization problem so as to discover optimal designs systematically.

5.1 Introduction



(a) Conventional DLD design cannot sort the cells since both cells zig-zag.



(b) A design with a triangular pillar cross section and narrower gaps can sort the cells.

Figure 5.1: *Top views of two DLD designs for deformability-based cell sorting. Flow is from left to right. Vertically aligned pillars (in gray) form columns and the pillar rows are tilted with respect to the flow axis. The red cell is stiffer than the blue cell. (a) The conventional design cannot sort the cells because it leads both to "zig-zag". (b) Solving a design optimization problem leads to a design with the same tilt angle of the pillar rows as in (a) but a triangular pillar cross section and narrower gaps that can sort the cells since the soft cell can move along the tilted pillar rows (i.e., in the displacement mode) and the stiff cell can zig-zag.*

There are a few studies on deformability-based sorting of RBCs in deep DLD devices [65, 179], including ours [77] (see also Chapter 4). Both ours and Henry et al. [65] considered devices with circular pillar cross sections and aimed to explain how cell sorting takes place. Zhang et al. [179] investigated cell dynamics for triangular,

square and diamond pillar cross sections. So, these recent studies are concerned with discovering cell dynamics in conventional DLD devices. Here we discover different DLD designs for efficient cell sorting. One of the difficulties is sorting cells with similar mechanical properties. That might either be impossible due to low sorting resolution or require long devices to induce sufficient vertical separation, which increases the process time. So, we need to design a device that is short but still capable of sorting such cells. Exhaustive search for that purpose is not practical because one needs to perform experiments or simulations to determine the cells' dynamics in every device. Also such computations are very expensive. How can we systematically design DLD devices for particular objectives and constraints? This is the main question we aim to address in this study.

Methodology. We presented our integral equation solver for 2D simulations of RBC flows and our high-fidelity DLD model in Chapters 2 and 4, respectively. Here, we use the same numerical scheme. Design parameters of a DLD device are pillar cross section (i.e., top view of pillars), tilt angle of pillar rows and center-to-center distances. These define a unique device. We fix the tilt angle and the center-to-center distances. So, the only design parameter is pillar cross section which we parameterize with uniform fifth-order B-splines. The objective function for the optimization problem assesses whether a design provides efficient cell sorting but quantifying this statement is not obvious. We discuss the choice of the objective function in Section 5.3.2. We solve the optimization problem using a stochastic optimization algorithm called the covariance matrix adaptation evolution strategy (CMA-ES) [57–60]. Evaluating the objective function requires

simulating cell flows through a DLD device. That is why it is infeasible to solve the optimization problem using our high-fidelity DLD model (HF-DLD). So, we propose a low-fidelity DLD model (LF-DLD) that has less number of unknowns than HF-DLD. We build HF-DLD once to obtain accurate boundary conditions for LF-DLD and perform the simulations using LF-DLD. Once the optimization is solved, the result is verified using HF-DLD. We also carefully analyze the sensitivity of cell dynamics to the perturbations in pillar cross sections using HF-DLD. We consider four sorting scenarios involving cells with similar viscosity contrast values. We compare the features of the optimal designs for these scenarios with the conventional ones which have equal gap sizes and circular, triangular, square and diamond cross sections.

Contributions. The main contribution of our study is to show that designing a DLD device can be posed as an optimization problem and solving it systematically discovers optimal designs as opposed to doing an exhaustive search. The optimal designs in the scenarios considered here are different than those in the literature and have cross sections similar to a triangle with a flat edge in the shift direction of the pillars (see Fig. 5.1). These designs are optimal in the sense that they provide large vertical separation between the cells after a few number of cell-pillar interactions. Thereby, they provide efficient cell sorting. The other contribution is a low-fidelity DLD model which reduces the computational cost of the simulations so that an optimization problem can be solved in a reasonable time (i.e., 3-4 days whereas it would take 15-16 days if the high-fidelity model was used.).

Limitations. Our simulations are in two dimensions. We have opted to use two-dimensional simulations since three-dimensional simulations for cell sorting via DLD can be quite expensive for optimization [151]. We consider only dilute suspensions. We do not allow changes in the resting size and shape of an RBC. We do not put any constraints on the pillar cross sections regarding the manufacturability such as symmetry of the cross sections. As a result of that, the optimal designs might have sharp and fine features. However, these designs can still help in designing an effective device with a simpler geometry in Section 5.6.

Related work. Let us review the literature on DLD designs. There are only a few studies considering pillar cross sections different than circular. Loutherback et al. [107] proposed a triangular pillar cross section and studied the effects of its size and orientation, vertex rounding on size-based sorting of rigid spherical particles. They found that the triangular pillar cross section shifts the flow in a gap towards the sharp vertex. This reduces the width of the stream adjacent to the sharp vertex compared to a circular cross section. So, for the same adjacent stream size a triangular pillar cross section allows using larger vertical gap size than a circular one, which not only increases the throughput but also reduces the risk of clogging. While a triangular cross section can be used to adjust the critical particle size for the separation of rigid particles, it cannot be straightforwardly used for deformability-based sorting of cells because it also affects cell dynamics which is not investigated in [107]. Our optimal designs have cross sections similar to a triangle and we investigate the effects of such cross sections on cell dynamics. In [108] the same group used such a design in an experimental study

to sort circulating tumor cells from whole blood and proved the advantages of their design. Al-Fandi et al. [3] aimed at proposing cross sections such that the cells deform only slightly and their dynamics can be predicted using the analytical DLD theory for rigid particles. They did not consider deformability-based sorting of the cells. The proposed cross section has an airfoil shape and results in a velocity field which does not deform cells as much as circular or diamond cross sections. Zeming et al. [177] aimed at designing a DLD device in which healthy RBCs displace. They suggested an I-shape cross section, which can induce rotational motion (tumbling) of the cells and hence lead them to stay out of the adjacent stream. They proved the effectiveness of the proposed design by conducting experiments. Ranjan et al. [149] experimentally studied the effects of various orientations of I-shape, T-shape and L-shape cross sections on the dynamics of rigid spherical particles and the cells. They concluded that cross sections with protrusions and grooves can induce tumbling of the cells and therefore, lead them to displace while keeping the rigid particles zig-zagging. Zhang et al. [179] conducted a numerical study to investigate cell dynamics in DLD devices with circular, square, diamond and triangular pillar cross sections. They stated that the prediction of the cell transit strongly depends on device geometry and structure. Therefore, they expected that new designs other than the circular pillar cross section can be useful for various objectives. All of these studies considered equal horizontal and vertical gap sizes. Recently, Zeming et al. [178] showed by conducting experiments that one can change particles' transport modes by using unequal gap sizes without changing the pillar cross section which was circular in particular. Although none of these studies investigated designs for sorting cells by their deformability, they contribute to our un-

derstanding of how various cross sections and unequal gap sizes affect cell dynamics. Our formulation results in cross sections that are different than the above studies and automatically adjusts both cross sections and gap sizes.

Organization of the Chapter. In Section 5.2, we introduce our low-fidelity DLD model. We propose an objective function and state the design optimization problem and explain how we solve it in Section 5.3. We, then, explain the numerical experiments we performed in Section 5.4 and discuss the results in Section 5.5. Finally, we illustrate how the solution of the optimization problem guides designing a DLD device with a simpler geometry (thus, possibly easier to manufacture) in Section 5.6.

5.2 Numerical Model

We, first, explain the pillar arrangement for an arbitrary cross section. The pillars are placed in a DLD device based on the smallest circumferential rectangle (see Fig. 5.2 for the schematic). Let h_{p} and w_{p} be the height and the width of this rectangle. Fluid flows in the x direction, which is the horizontal direction. We denote the horizontal and vertical gap sizes between the rectangles with G_x and G_y, respectively. The gap sizes are also the minimum spacings between the pillars. The horizontal and the vertical center-to-center distances between the rectangles become $\lambda_x = G_x + w_{\mathrm{p}}$ and $\lambda_y = G_y + h_{\mathrm{p}}$. Each column is shifted in the vertical direction with respect to the previous column by $\Delta\lambda$ which is defined as $\Delta\lambda = \tan(\theta)\lambda_x$ for the tilt angle of the pillar rows θ. The tilted pillar row arrangement divides the flow in a vertical gap into a several streams carrying equal flux. After every vertical gap the stream adjacent to a

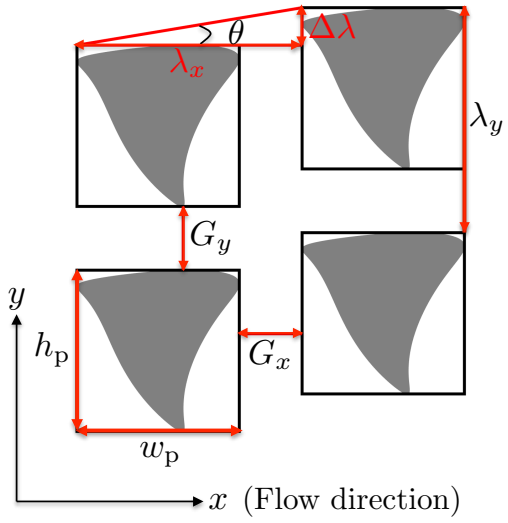


Figure 5.2: *Top view of the lattice of pillars with arbitrary cross sections (in gray) in a DLD device. G_x, G_y denote the gap sizes, and θ is the tilt angle of the pillar rows. Fluid flows in the x direction (horizontal) and each pillar column is shifted in the y direction (vertical) by $\Delta\lambda$ with respect to a previous column. We set the gap sizes such that they are the spacings between the circumferential rectangles. h_p and w_p are the height and the width of the rectangle. The center-to-center distances in the x and the y directions between the rectangles are $\lambda_x = G_x + w_\mathrm{p}$ and $\lambda_y = G_y + h_\mathrm{p}$, respectively.*

pillar, the adjacent stream swaps a lane by moving downwards. The number of these streams is $\lceil n \rceil$, where

$$n = \frac{\lambda_y}{\Delta\lambda}. \tag{5.1}$$

n is also referred to as the number of columns in a period of the device if n is an integer. "Period" sets a length scale in which the column arrangement is exactly repeated. If n is not an integer, the column arrangement does not repeat itself.

Actual DLD devices usually consist of $\mathcal{O}(10)$ tilted rows and $\mathcal{O}(100)$ columns, which results in $\mathcal{O}(1000)$ pillars in a device. Performing simulations of cell sorting in a whole device is computationally very expensive even for a single simulation, let alone

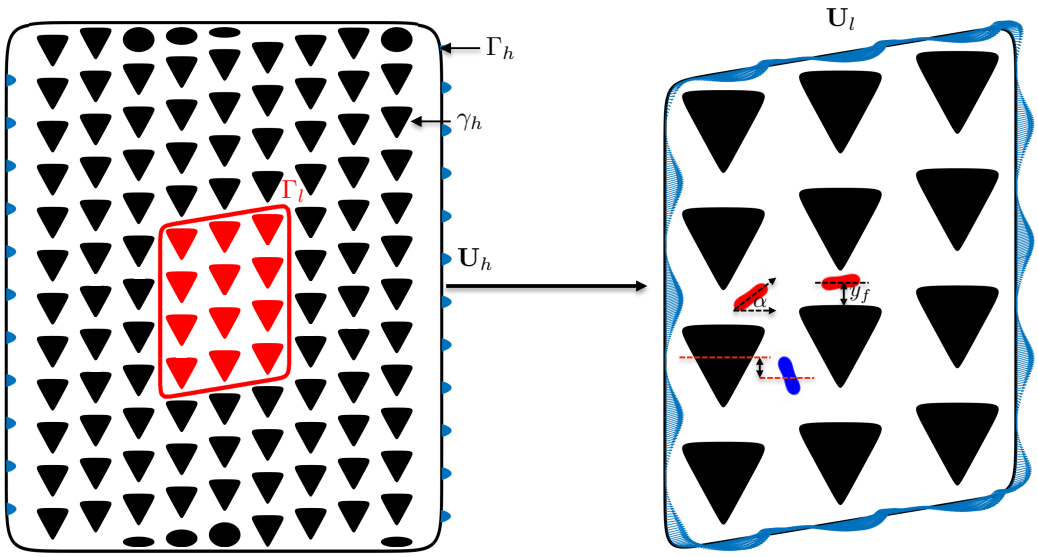
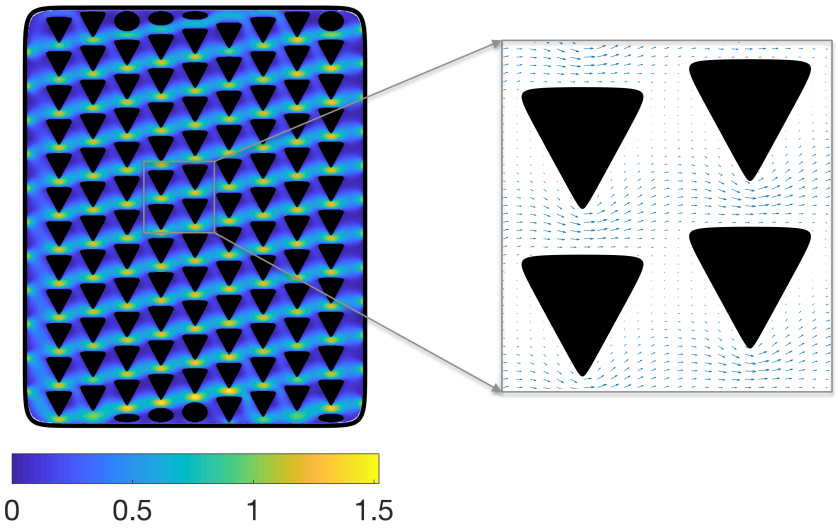


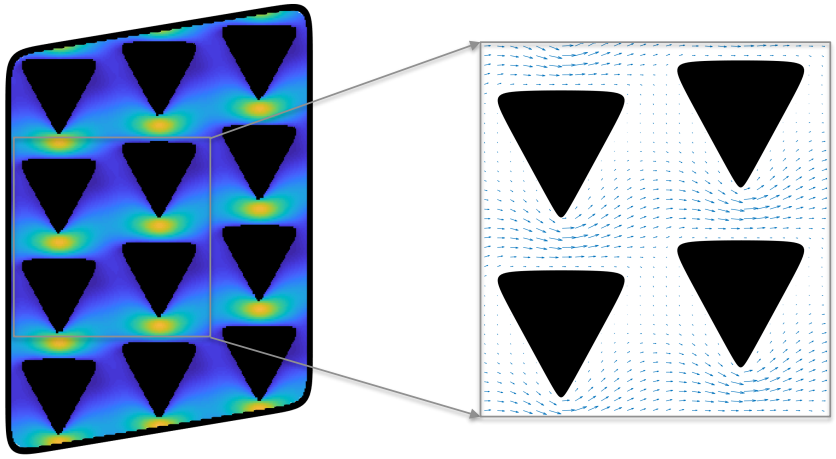
Figure 5.3: *High-fidelity model (on the left) results in large number of unknowns, which renders its use for optimization impractical. We develop a low-fidelity model (on the right) which is constructed as follows. Side walls in the high-fidelity model pass through where the vertical gap size between the pillars is minimum (i.e., G_y) on the imaginary columns on the left and on the right. We assign a parabolic velocity $\mathbf{U}_h$ on the imaginary gaps on Γ_h (blue arrows in the left figure) and zero velocity on γ_h. Then, we solve (5.2) to obtain the density ζ_h on $\Gamma = \Gamma_h \cup \gamma_h$. Finally, we compute the velocity $\mathbf{U}_l$ at $\mathbf{x} \in \Gamma_l$ due to the density ζ_h using (5.3). We impose $\mathbf{U}_l$ as a Dirichlet boundary condition when simulating cell flows using the low-fidelity model as in the right figure. y_f on the right figure is the vertical distance between the displacing cellâĂŹs center and the top of the pillar underneath it at the end of the simulation. α is the inclination angle of the cell (see Section 2.5.1 for the definition of the inclination angle).*

for optimization. We introduced a high-fidelity approximate DLD model in Section 4.2.1. Here, we modify it for arbitrary pillar cross section and gap sizes. We also propose a low-fidelity model. Key parameters in these models are the numbers of rows (width) and columns (length) and the boundary conditions applied on the exterior wall. To describe the length we use the notion of the "period", which we introduced in (5.1), and $\lceil n \rceil$ is the number of columns per period. Let us explain these models.

High-fidelity model (HF-DLD). In Section 4.2.1 we numerically determined that wall effects are negligible if we use 12 rows and $\lceil 1.5n \rceil$ columns. However, here we observed that these numbers of rows and columns depend on the gap sizes and the pillar cross sections. That is why in this study HF-DLD has 12 rows and 9 columns of pillars as in the left figure in Fig. 5.3. Since wall effects are minimum in the middle of the device, we initialize a cell there. As the cell travels and reaches to the next column, we translate it back to the previous column. This trick is possible since, unlike in Chapter 4, we are interested in a single cell in this study. So simulations take place between only two columns. Convergence studies for the cell trajectories showed that the wall effects are negligible for various pillar cross sections in HF-DLD. Using this model we can simulate a single cell for an arbitrary number of periods with a much smaller cost. Initial position and orientation of a cell have to respect the physics of cell flows in DLD. Displacing cells have asymptotic periodic motion with a certain distance to pillars in a gap and positive inclination angles (see Section 4.4.2). This certain distance depends on the cell's deformability and the flow field. Zig-zagging cells get much closer to pillars and have negative inclination angles. In order to minimize the uncertainty in the initial position and orientation of a cell, we initialize it in the middle of a vertical gap with zero inclination angle. Although the flow is not symmetric in the gap for arbitrary pillar cross sections, we still impose a symmetric parabolic velocity as a boundary condition ($\mathbf{U}_h$ in the left figure in Fig. 5.3). While this introduces an error, it is negligible in the middle of the device.



(a) Velocity in HF-DLD



(b) Velocity in LF-DLD

Figure 5.4: *Velocity magnitude and field in HF-DLD (a) and LF-DLD (b) without RBCs for a triangular pillar cross section. The gap sizes are* $G_x = G_y = 7.5\mu m$. *The dimensions of the triangles are* $h_p = w_p = 17.5\mu m$.

Low-fidelity model (LF-DLD). Although HF-DLD is much cheaper than a whole device with hundreds of columns, it is still too expensive for optimization. To further

reduce the cost we introduce a low-fidelity model that has four rows and three columns along with an exterior wall (see the right figure in Fig. 5.3). To make LF-DLD more realistic we place the exterior wall as if it passes in the middle of the gaps between the pillars (i.e., Γ_l in the left figure in Fig. 5.3). Then, we use the velocity field from HF-DLD (without RBCs) as a boundary condition for the exterior wall in LF-DLD. We do this as follows (see also Fig. 5.3 for the schematic).

- Let Γ_h and Γ_l denote the exterior walls in HF-DLD and LF-DLD, respectively. Also γ_h denotes the boundary of the pillars in HF-DLD.

- We impose $\mathbf{U}_h$ as the velocity on Γ_h and zero velocity on γ_h. We solve the second-kind Fredholm integral equation (5.2) for the hydrodynamic density ζ_h on the boundary $\Gamma = \Gamma_h \cup \gamma_h$ [137],

$$\mathbf{U}_h(\mathbf{x}) = -\frac{1}{2}\zeta_h(\mathbf{x}) + \frac{1}{\pi}\int_\Gamma \frac{\mathbf{r}\cdot\mathbf{n}}{\|\mathbf{r}\|^2}\frac{\mathbf{r}\otimes\mathbf{r}}{\|\mathbf{r}\|^2}\zeta_h(\mathbf{y})ds_y, \quad \mathbf{x}\in\Gamma \tag{5.2}$$

  where $\mathbf{r} = \mathbf{x} - \mathbf{y}$.

- Using (5.3) we compute the velocity $\mathbf{U}_l$ (see right figure in Fig. 5.3) at the discretization points $\mathbf{x}\in\Gamma_l$ due to the density $\zeta_h(\mathbf{y})$, $\mathbf{y}\in\Gamma$ [137],

$$\mathbf{U}_l(\mathbf{x}) = \frac{1}{\pi}\int_\Gamma \frac{\mathbf{r}\cdot\mathbf{n}}{\|\mathbf{r}\|^2}\frac{\mathbf{r}\otimes\mathbf{r}}{\|\mathbf{r}\|^2}\zeta_h(\mathbf{y})ds_y, \quad \mathbf{y}\in\Gamma. \tag{5.3}$$

The density ζ_l due to $\mathbf{U}_l$ represents the hydrodynamic sources outside Γ_l in HF-DLD. Hence, LF-DLD and HF-DLD give the same velocity at any point in Γ_l in the absence of RBCs. See Fig. 5.4 for the velocity magnitudes and fields for HF-DLD and LF-DLD for a triangular pillar cross section. The presence of the cells in HF-DLD would

perturb the velocity $\mathbf{U}_l$ if it was computed at every time step. For a number pillar cross sections, we measured the space-time average of the perturbation and found that it does not exceed 5%. Therefore, we consider LF-DLD reliable for optimization. So to be clear, this calculation needs to be repeated whenever the pillar cross section changes but it does not need to be repeated within the calculation of the RBC trajectories.

Remark. Since LF-DLD is built from HF-DLD for every pillar cross section, it is important to do the calculation efficiently (in terms of time and storage). When computing ζ_h, the discretization of (5.2) at a set of collocation points on $\Gamma_h \cup \gamma_h$ and replacing the integral with the trapezoid rule result in a dense linear system with $\mathcal{O}(10^4)$ unknowns. We iteratively solve this system matrix-free with GMRES. In order to accelerate the solver, the matrix-vector multiplication is done in linear time with fast multipole method [55]. We reduce the number of GMRES iterations by applying a block-diagonal preconditioner. Here, each block corresponds to the completed double-layer potential for each solid boundary.

5.3 Design Optimization Problem

For given two different viscosity contrast values, we seek a DLD design that sorts the cells by their viscosity contrasts. In Section 5.3.1 we explain the device parameterization. Then, we state the optimization problem and propose an objective function in Section 5.3.2. See Section 4.2.4 for the dimensionless numbers used in this chapter.

5.3.1 Device Parameterization for Optimization

The pillar cross section, center-to-center distances between the circumferential rectangles, and the tilt angle of the pillar rows are free design parameters. We seek designs that are small and result in certain vertical displacement between the cells, which provides efficient sorting. This amounts to fixing the tilt angle and the device size. In the optimization problem we fix the center-to-center distances and the tilt angle since the reported experiments [115] and the numerical studies on sorting RBCs using DLD give information about the pillar arrangement, not the device size. That is, λ_x, λ_y and θ in Fig. 5.2 are fixed. Thus, the only free design parameter is pillar cross section which we parameterize using uniform fifth-order B-splines (see Section 5.7 for the details). For any cross section with size h_{p} and w_{p}, we deduce the horizontal and the vertical gap sizes from the center-to-center distances between the circumferential rectangles, i.e., $G_x = \lambda_x - w_{\mathrm{p}}$ and $G_y = \lambda_y - h_{\mathrm{p}}$. With that we have all the design parameters to construct a DLD device. So to be clear, a DLD design involves a pillar cross section, center-to-center distances (or gap sizes) and a tilt angle.

In the optimization, we have to make sure that the velocity fields between different optimization iterations are consistent. To this end, for each proposed pillar cross section, we adjust the velocity boundary conditions so that the imposed total flow rate is the same for all designs. We call DLD designs invalid if they have self-intersecting or rough cross sections. Additionally, small gap sizes lead to large velocity and large pressure drop and increase the risk of clogging. The experimental studies reported so far have used gap sizes greater than $7\mu m$ [115]. Here we set the minimum gap size allowed to $7\mu m$ and call a design invalid if it has gap sizes smaller than that.

We decide whether a cross section is rough using the decay of the energy spectrum of the cross section. If the high-frequency energy exceeds the low-frequency energy, then we consider that cross section rough. Those invalid configurations are rejected by assigning a high default objective function value.

Remark. It is also possible to optimize the tilt angle and the separation between pillars in addition to the pillar cross section. This does not pose any numerical challenge and it would be even easier to find a design that can sort the cells. Here we aim at optimizing designs for more difficult cases (i.e., under constraints and for very similar cells).

5.3.2 Optimization Problem

Given the viscosity contrast values of two cells $(\nu_1,\ \nu_2)$, the center-to-center distance between the circumferential rectangles $(\lambda_x,\ \lambda_y)$ and the tilt angle of the pillar rows θ, we find an optimal design by (i) choosing a design, (ii) performing simulations of the cells using LF-DLD, and (iii) then evaluating the objective function to compare dynamics and decide if the design is acceptable. In order to choose a design systematically we use the *covariance matrix adaptation-evolutionary strategy (CMA-ES)* [57–60] as an optimization algorithm (see Section 5.3.3 for a brief description of the algorithm). It samples designs from a Gaussian distribution which is updated based on the evaluations of the objective function for the sampled designs in the course of the optimization. The only adjustable parameter of the CMA-ES is the number of samples used in an iteration to update the Gaussian distribution. We set this number to 32

and using a 32-core processor we perform embarrassingly parallel cell simulations to evaluate the objective function. This enables fast solution of the optimization problem. We terminate the iterations when the overall standard deviation decreases below 0.05 or becomes stationary for a few iterations.

We now propose an objective function which compares cell dynamics and assesses whether a design provides efficient cell sorting. Let us introduce the following qualitative definitions that characterize the efficiency of the device: If both cells are sorted (one displaces one zig-zags) then a device is in "separation mode". Otherwise the device is in "no-separation mode". In order to numerically determine whether a cell displaces or zig-zags we run cell simulations until the cell travels one period of a device, i.e., $\lceil n \rceil$ columns. Recall that the simulations take place between the first two columns in LF-DLD. If the cell's center is below the center of the pillar above which it is initialized, we call it zig-zagging. For instance, the blue cell in the right figure in Fig. 5.3 zig-zags. Otherwise, we call it displacing. We require the objective function to (i) give always smaller values for designs in separation mode than for those in no-separation mode because only separation mode is desirable, (ii) quantify the difference in cell dynamics for both modes, i.e., distinguish between two designs in separation mode and similarly for those in no-separation mode, and (iii) in particular, for distinguishing devices in separation mode, decrease when the displacing cell migrates more in the vertical direction or the zig-zagging cell zig-zags earlier because this increases the net vertical separation between the cells and hence provides efficient sorting by reducing

the process time. Based on these considerations, we define the objective function

$$f = \begin{cases} -C_1 \frac{y_f}{G_y} + C_2 \frac{n_{zz}}{\lceil n \rceil}, & \text{for separation mode,} \\ C_3 - |\frac{\Delta y_f}{G_y}| & \text{for no-separation mode where both cells displace,} \\ C_3 - |\frac{\Delta n_{zz}}{\lceil n \rceil}| & \text{for no-separation mode where both cells zig-zag,} \end{cases} \tag{5.4}$$

where C_1, C_2, C_3 are positive constant coefficients, y_f is the vertical distance between the displacing cell's center and the top of the pillar underneath it at the end of the simulation (see the right figure in Fig. 5.3), n_{zz} is the number of columns after which the zig-zagging cell zig-zags for the first time and Δq stands for the difference between the values quantity q of one cell and the other cell. Let us interpret (5.4).

- We normalize the vertical displacement of the displacing cell y_f by the vertical gap size G_y, $\frac{y_f}{G_y}$, which tells how much the cell migrates away from a pillar. As the cell goes away from the pillar to the middle of the vertical gap, it travels faster and the sorting becomes quicker.

- For separation mode, f is the difference between the normalized vertical displacement of the displacing cell ($\frac{y_f}{G_y}$) and the normalized number of columns after which the zig-zagging cell zig-zags ($\frac{n_{zz}}{\lceil n \rceil}$). It decreases if the displacing cell migrates more in the vertical direction or the zig-zagging cell zig-zags after a less number of columns. Therefore, a design that results in smaller f provides more efficient cell sorting.

- The coefficients C_1 and C_2 assign different importance on the displacing and the zig-zagging cells, respectively. Our simulations showed that the value of $\frac{y_f}{G_y}$ is usually about 0.2. However, the ratio $\frac{n_{zz}}{\lceil n \rceil}$ is close to 1. We make these ratios

comparable using $C_1 = 1$ and $C_2 = 0.2$. This allows the objective function to decrease by the same amount due to more vertical displacement of the displacing cell and earlier zig-zagging of the zig-zagging cell.

- For no-separation mode, f quantifies how different cell dynamics are. We quantify this difference as either the difference in the vertical displacement of the displacing cells or the difference in the number of columns after which the zig-zag occurs. The greater this difference is, the more possible it is to separate cells. So we want to maximize it. That is, we minimize $-|\frac{\Delta y_f}{G_y}|$ for the displacing cells and $-|\frac{\Delta n_{zz}}{\lceil n \rceil}|$ for the zig-zagging cells.

- The coefficient C_3 is chosen to ensure that separation mode results in a smaller objective function than no-separation mode. Running a few simulations we found that $C_3 = 10$ is sufficient.

Overall, we state the design optimization problem as follows: Minimize $f(c)$ such that $G_x(c), G_y(c) \geq G_{\min} = 7\mu m$ and $\mathbf{X}(c)$ is smooth and not self-intersecting, where f is (5.4), c is the coordinates of the B-spline control points, and $\mathbf{X}$ is a pillar cross section.

Remark. We need the objective function to be discontinuous for the following reason. We are interested in designs in separation mode only, however, we want those in no-separation mode to inform the optimizer for faster convergence. That is why we do not reject designs in no-separation mode by assigning a high default objective function value. Instead, we make the objective function continuously change among the designs

in separation mode and among those in no-separation mode but has a jump between these modes. As a result of that, it can distinguish two designs in separation mode and in no-separation mode. One can also use an overall measure of vertical separation as an objective function, e.g., the difference in the angle at which the cells migrate on average. Such an objective function is insensitive to how much a displacing cell migrates from the pillar and how early a zig-zagging cell zig-zags. However, we seek designs that result in more migration of the displacing cell and earlier zig-zag of the zig-zagging cell, which provides fast sorting. That is why we decided to use (5.4) as an objective function.

5.3.3 Optimization Algorithm: CMA-ES

We present the CMA-ES algorithm in Algorithm 6. The notation is limited to this section and does not refer to any previous use of the symbols. The input to the CMA-ES is population size λ, initial mean $\mathbf{m}^{(0)}$ and overall standard deviation $\sigma^{(0)}$ for the parameters. The only input parameter to be tuned is the population size λ. While small population sizes lead to faster convergence, large population sizes avoid being stuck in local optima. Since sampling offspring and evaluating an objective function are embarrassingly parallel, we parallelize this section of the code (the for loop in Algorithm 6). Considering our computational sources and time to solution, we chose $\lambda = 32$. The CMA-ES is a black-box algorithm, i.e., the other parameters in Algorithm 6 are already set heuristically. These parameters are the weights $w_{i=1,\cdots,\lambda}$ such that $\sum_{i=1}^{\lambda} w_i = 1$, the number of offspring used to update mean and covariance matrix $\mu = \lambda/2$, learning parameters $c_\sigma, d_\sigma, c_{cov}, c_1, c_\mu$. The parameter μ_{eff} is the variance

effective selection mass and set to $\mu_{\text{eff}} = 1/\sum_{i=1}^{\mu} w_i^2$. The CMA-ES form in Algorithm 6 is called rank-μ and weighted recombination since it uses the best μ offspring and the weighted sum in updating the mean and the covariance matrix.

Algorithm 6 CMA − ES: rank-μ and weighted recombination form

Require: $\lambda, \mathbf{m}^{(0)}, \sigma^{(0)}$
  Fixed parameters: $w_{i=1,\cdots,\lambda}, \mu, c_\sigma, d_\sigma, c_{cov}, c_1, c_\mu, \mu_{\text{eff}}$
  Initialize: $\mathbf{p}_\sigma = \mathbf{0}, \mathbf{p}_c = \mathbf{0}, \mathbf{C} = \mathbf{I}, g = 0$
  while (not termination criteria) **do**
    $g \leftarrow g + 1$
    for $i = 1, \cdots, \lambda$ **do** — λ: number of offspring in a generation
      $\mathbf{x}_k^{(g+1)} \sim \mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2 \mathbf{C}^{(g)})$ — Sample offspring from a normal distribution
      $f_k^{(g+1)} \leftarrow f(\mathbf{x}_k^{(g+1)})$ — Evaluate offspring
    end for
    $(f_{i:\lambda}^{(g+1)})_{i=1}^{\lambda} = \texttt{sort}(f_k^{(g+1)}, \texttt{'ascend'})$ — Sort function evaluations in ascending order
    // Let $\mathbf{x}_{i:\lambda}^{(g+1)}$ be defined such that $f(\mathbf{x}_{i:\lambda}^{(g+1)}) = f_{i:\lambda}^{(g+1)}$
    $\mathbf{m}^{(g+1)} \leftarrow \mathbf{m}^{(g)} + c_\mu \sum_{i=1}^{\mu} w_i (\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)})$ — Select the best μ offspring and update mean
    $\mathbf{p}_\sigma^{(g+1)} \leftarrow (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}\mathbf{C}^{-1/2} \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}^{(g)}$ — Adjust step size
    $\sigma^{(g+1)} \leftarrow \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\mathrm{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right)$ — E(·) indicates expected value
    $\mathbf{p}_c^{(g+1)} \leftarrow (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}^{(g)}$
    $\mathbf{C}^{(g+1)} \leftarrow (1 - c_{cov})\mathbf{C}^{(g)} + c_1 \mathbf{p}_c^{(g+1)} (\mathbf{p}_c^{(g+1)})^T + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}^{(g)} (\mathbf{y}_{i:\lambda}^{(g)})^T$ — Update covariance matrix
  end while

5.4 Numerical Experiments

We considered four sorting scenarios with different viscosity contrast values ν_1, ν_2 and tilt angles θ. The details are as follows.

- **Scenario 1**: ($\nu_1 = 5$, $\nu_2 = 10$) and $\theta = 0.17$ rad. A healthy RBC has viscosity contrast $\nu \in [4, 6]$ [120]. So we consider a healthy cell and a cell slightly stiffer than that. In this and the fourth scenarios, the viscosity contrasts are the closest compared to the other scenarios. So cell dynamics are the most similar and

hence it is difficult to separate these cells. The critical viscosity contrast value for separation must be between 5 and 10. The purpose of this experiment is to see if it is possible to design a device to sort cells with very similar dynamics.

- **Scenario 2**: ($\nu_1 = 4,\ \nu_2 = 10$) and $\theta = 0.17$ rad. We keep the viscosity contrast of the stiff cell the same as the previous scenario and make the soft cell a little softer. If there is an optimal device for the previous scenario, it sorts the cells in this scenario as well since that device must have the critical viscosity contrast value for separation between 5 and 10. Here we aim to investigate how much the optimal design for the previous scenario changes due to a slight change in the cells' viscosity contrasts.

- **Scenario 3**: ($\nu_1 = 5,\ \nu_2 = 50$) and $\theta = 0.17$ rad. This is the easiest scenario since the viscosity contrast of the stiff cell is 10 times greater than the soft cell. So optimal devices for the previous scenarios can sort the cells in this scenario as well. Here the stiff cell cannot migrate as much as the stiff cells in the previous scenarios due to its greater viscosity contrast value. Our goal is to observe how this fact affects the optimal design.

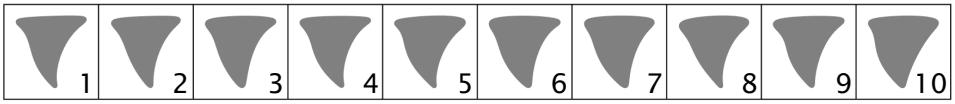
- **Scenario 4**: ($\nu_1 = 5,\ \nu_2 = 10$) and $\theta = 0.2$ rad. We consider the same cells as in scenario 1 but we set the tilt angle to a greater value than in scenario 1. The size of the adjacent stream increases with the tilt angle and hence the soft cell has to migrate more to displace. Separating the cells is more difficult in this scenario than in scenario 1. So we want to inquire if it is still possible to find a design to separate these cells.

In all scenarios we set $\lambda_x = \lambda_y = 25\mu m$ because these are the typical dimensions for a DLD device for rigid or deformable particles [115]. We impose the horizontal flux of $5225(\mu m)^2/s$ in a vertical gap, which sets the capillary number to $C_a = 3.75$. We set the tilt angle of the pillar rows to $\theta = 0.17$ rad in the first three scenarios and to $\theta = 0.2$ rad in the last scenario. These tilt angles are relatively large compared to the typical angles used in the real devices [115]. We have chosen large angles because they are more challenging (separation is harder to achieve) but also more desirable (if separation is possible, it is faster since shorter devices can be used).

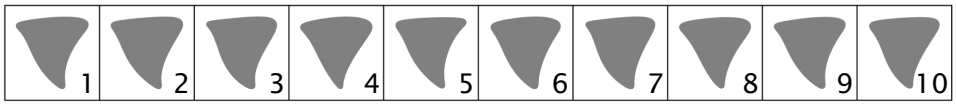
We use the same initial guess for all scenarios. The initial guess has a circular cross section with the diameter $15\mu m$ and the gap sizes are $G_x = G_y = 10\mu m$. For all the scenarios that we described above this design does not sort the cells (i.e., in no-separation mode) (see Section 4.3.3). All the numerical experiments are done in MATLAB on 32-core 2.60GHz Intel Xeon processor with 256GB memory.

5.5 Results and Discussions

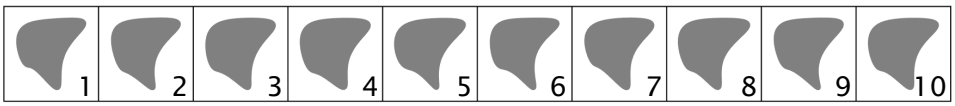
After a number of iterations the optimization algorithm was able to find optimal designs in separation mode for all the scenarios. These designs are optimal only for the scenarios, the objective function (5.4) we defined and under the size constraints. We first present and discuss the optimal DLD designs in Section 5.5.1. Then we show the results of the high-fidelity model simulations using the initial guess and the optimal designs in Section 5.5.2.



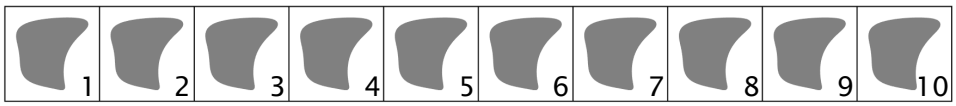
(a) Scenario 1: $(v_1, v_2) = (5, 10)$ and $\theta = 0.17$ rad



(b) Scenario 2: $(v_1, v_2) = (4, 10)$ and $\theta = 0.17$ rad



(c) Scenario 3: $(v_1, v_2) = (5, 50)$ and $\theta = 0.17$ rad



(c) Scenario 4: $(v_1, v_2) = (5, 10)$ and $\theta = 0.2$ rad

Figure 5.5: *Optimal cross sections for all scenarios. See Table 5.1 for the cell dynamics and the gap sizes in these designs.*

5.5.1 Optimal Designs

For each sorting scenario we present the pillar cross sections for the ten most optimal designs in Fig. 5.5. Here the cross sections are ordered such that the first one results in the smallest objective function value (i.e., the most efficient sorting) and the squares around the cross sections have dimensions $\lambda_x = \lambda_y = 25\mu m$. See Section 5.7 for the coordinates of the control points for B-spline curves and reproduction of the cross sections. We also tabulate the gap sizes G_x, G_y in these designs, the vertical displacements of the displacing cells y_f/G_y and the numbers of columns after which zig-zagging occurs n_{zz} in Table 5.1. The designs delivering the smallest objective function

value result in either earlier zig-zag of the stiff cells or larger vertical migration of the soft cells from the pillars, which leads to larger vertical separation of the cells. So the small objective function values correlate with the large vertical separation of the cells. We want to compare the critical viscosity contrast values for separation for the optimal designs in the first three scenarios as well. For that purpose, we perform simulations of the cells with $\nu = (4, 5, 10, 50)$ in the optimal designs and present the cells' transport modes in the left figure in Fig. 5.6. We, now, discuss these results based on the questions raised in Section 5.4.

The initial guess consists of cylindrical pillars and has gap sizes $G_x = G_y = 10\mu m$. In all sorting scenarios both soft and stiff cells zig-zag in this initial design. A design in separation mode must lead one of the cells to displace by inducing either more vertical migration and a thinner adjacent stream or inducing tumbling motion while keeping the other cell zig-zagging. The optimal designs we found lead the softer cells to displace and the stiffer ones to zig-zag in all scenarios. These designs have smaller gap sizes and hence thinner adjacent streams than the initial guess. If the pillar cross sections had remained circular, this might be sufficient to separate the cells. In fact, we performed an exhaustive search to find gap sizes that induce separation for the circular pillars (see the right figure in Fig. 5.6 for the results). The design with circular pillars and gap sizes $G_x = G_y = 7.5\mu m$ can sort the cells in the first three scenarios. However, this design is not among the optimal designs we found because the softer cells cannot migrate as much as they do in the optimal designs. So it has a greater objective function value (i.e., less efficient sorting) than the optimal designs.

Table 5.1: *We tabulate the gap sizes G_x, G_y, the vertical displacements of the displacing cells y_f/G_y and the numbers of columns after which zig-zagging occurs n_{zz} for the optimal designs that have the cross sections in Fig. 5.5.*

Scenario 1										
Design	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$G_x(\mu m)$	7.5	7.8	7.4	7.3	7.3	7.3	7.6	7.7	7.3	7.5
$G_y(\mu m)$	7.0	7.1	7.1	7.2	7.0	7.1	7.1	7.0	7.2	7.1
y_f/G_y	0.356	0.380	0.378	0.376	0.349	0.372	0.371	0.370	0.369	0.344
n_{zz}	3	4	4	4	3	4	4	4	4	3
Scenario 2										
Design	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$G_x(\mu m)$	7.0	7.1	7.5	7.1	7.4	7.7	7.8	7.3	7.3	7.6
$G_y(\mu m)$	7.1	7.0	7.0	7.0	7.4	7.0	7.0	7.4	7.2	7.3
y_f/G_y	0.377	0.402	0.375	0.374	0.398	0.370	0.370	0.394	0.369	0.394
n_{zz}	3	4	3	3	4	3	3	4	3	4
Scenario 3										
Design	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$G_x(\mu m)$	7.7	7.8	7.6	7.5	7.3	7.8	7.7	7.1	7.7	7.5
$G_y(\mu m)$	7.1	7.0	7.1	7.2	7.2	7.1	7.0	7.1	7.1	7.2
y_f/G_y	0.367	0.364	0.363	0.362	0.362	0.361	0.361	0.360	0.360	0.359
n_{zz}	2	2	2	2	2	2	2	2	2	2
Scenario 4										
Design	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$G_x(\mu m)$	7.1	7.0	7.0	7.1	7.1	7.2	7.1	7.1	7.2	7.2
$G_y(\mu m)$	7.0	7.1	7.1	7.0	7.0	7.0	7.0	7.0	7.0	7.0
y_f/G_y	0.371	0.370	0.370	0.370	0.370	0.370	0.370	0.370	0.369	0.369
n_{zz}	3	3	3	3	3	3	3	3	3	3

Let's try to understand why the optimal designs perform better. They have cross sections with a flat edge at the top and a sharp vertex at the bottom. Such cross sections result in an asymmetric flow in the vertical gap as opposed to a symmetric flow induced by a circular cross section [107]. The flow is shifted towards the sharp vertex (e.g., see Fig. 5.4(b)). The flow is from left to right for the optimal cross sections in Fig. 5.5. Therefore, the flow is shifted upwards. This has two consequences: It results in (i) stronger vertical migration due to larger positive flow curvature and (ii) a thicker adjacent stream compared to a circular cross section with the same gap sizes. Vertical migration is desirable since it can lead cells to displace. An overly thick adjacent stream can be problematic since it can prevent softer cells from displacing. The optimal designs avoid that by having narrower vertical gaps, which decreases the width of the adjacent stream. In order to illustrate that the flow shifted upwards induces more migration, consider the following example. We turn the optimal cross section for scenario 1 upside down, i.e., the sharp vertex is at the top and the flat edge is at the bottom. This results in a thinner adjacent stream since the flow is shifted downwards. We then perform a simulation of the cells in scenario 1 in this configuration. This design is also capable of cell sorting: While the stiff cell zig-zags, the soft cell displaces. However, the soft cell migrates less in this design ($y_f/G_y = 0.12$) than in the optimal design ($y_f/G_y = 0.36$) due to the flow shifted downwards. So the designs we found are optimal since they not only induce cell sorting but also lead the soft cell to displace vertically more than any other possible designs. This discussion also shows that adjusting the width of the adjacent stream does not guarantee separation of cells unlike rigid particles, one has to consider cell migration as well.

Comparing the optimal designs for the first two scenarios illustrates that a small change in the viscosity contrast value leads to visible but not significant changes in the optimal designs. The stiff cells are the same in these scenarios and have $\nu_2 = 10$. The softer cells have $\nu_1 = 5$ and $\nu_1 = 4$ in scenarios 1 and 2, respectively. The optimal designs for scenario 2 (see Fig. 5.5(b)) have cross sections similar to those for scenario 1 (see Fig. 5.5(a)).

In scenario 3, the soft cell is the same as in scenario 1 but the stiff cell is much stiffer, so it migrates less. The optimal designs for scenario 3 in Fig. 5.5(c) are quite different than scenario 1. Although the cross sections have a flat edge at the top and a sharp vertex at the bottom, they do not resemble a triangle anymore, unlike those in the first two scenarios. Additionally, while the vertical gap sizes are similar, the horizontal gap sizes are greater. The left figure in Fig. 5.6 shows that the critical viscosity contrast is higher in the optimal design for scenario 3 than for scenario 1. This is because the optimal design for scenario 3 induces more cell migration. Although the optimal design for scenario 3 leads the softer cell to migrate more, it does not qualify as one of the optimal designs for scenario 1 since it leads the stiffer cell to displace as well.

In scenario 4, the cells have the same viscosity contrast values as in scenario 1 but the tilt angle is larger. Increasing the tilt angle increases the width of the adjacent stream and leads cells to zig-zag for a wide range of viscosity contrasts [77]. That is why in order for the softer cell to displace cell migration must be stronger than scenario 1. The optimal design for scenario 4 is different than those for the other scenarios. It can induce more migration. When the optimal cross section for scenario 1 is used with the same tilt angle in scenario 4, separation is still possible but the soft cell migrates

less.

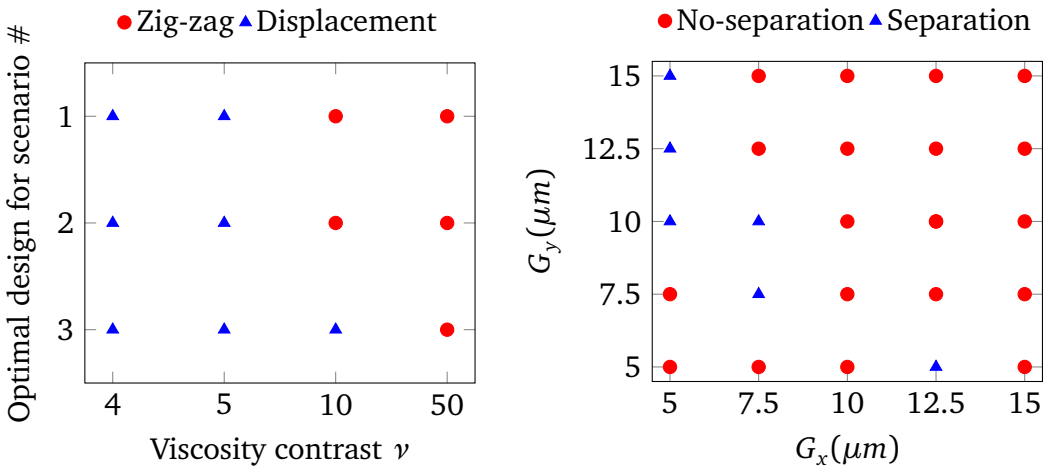


Figure 5.6: *On the left: phase diagram for the transport modes of the cells in the optimal designs for the scenarios 1, 2 and 3 as a function of viscosity contrast ν. Displacing and zig-zagging cells are demonstrated with a blue triangle and a red circle, respectively. On the right: phase diagram for sorting cells with the viscosity contrasts $\nu_1 = 5$ and $\nu_2 = 10$ using cylindrical pillars as a function of the gap sizes G_x and G_y. We indicate pairs of gap sizes leading to separation with a blue triangle and to no-separation with a red circle.*

Remark. It is also possible to have two cells sorted by using cylindrical pillars and changing the gap sizes only [178]. For the first scenario, we perform an exhaustive search to find the gap sizes which result in separation using cylindrical pillars with a diameter of $15\mu m$. We limit the search to the interval $(G_x, G_y) \in [5\mu m, 15\mu m]$. Unlike the optimization problems, we did not enforce $\lambda_x = \lambda_y = 25\mu m$. It turns out that separation is possible for several pairs of gap sizes shown in Fig. 5.6. However, notice that we do not allow any gap size below $7\mu m$ in the optimization. The design with equal spacings $G_x = G_y = 7.5\mu m$ could be chosen by the optimizer but it does not deliver as

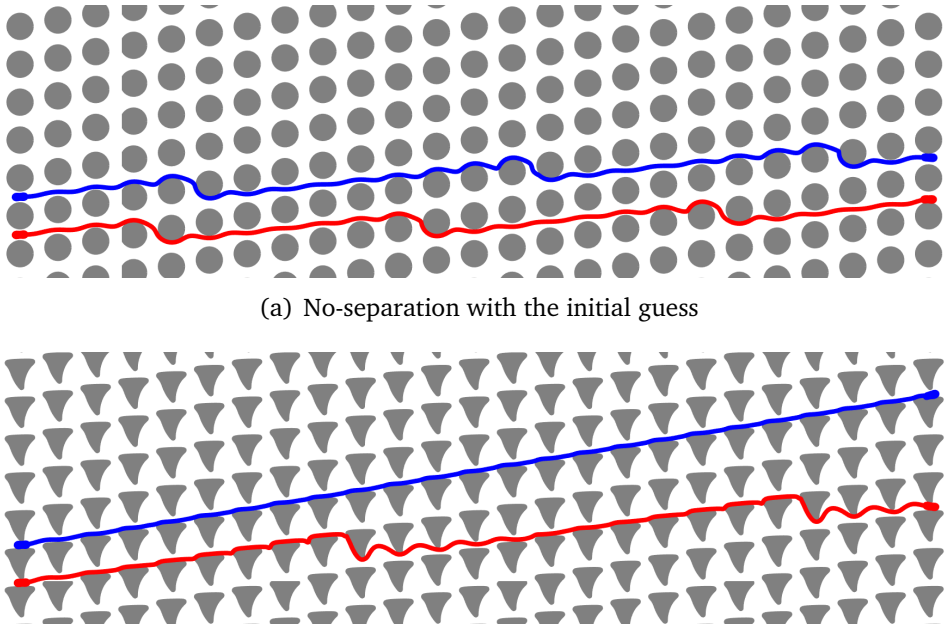
small objective function value as the optimal designs. That is, it does not provide sorting as efficient as the optimal designs do. The other design with $(G_x, G_y) = (7.5, 10)\mu m$ was not allowed in the optimization problem due to the fixed values of $\lambda_x = 25\mu m$ and $\lambda_y = 25\mu m$.

5.5.2 High-fidelity DLD Simulations

Recall that in the optimization solve, we used the low-fidelity model. But do the designs work in the high-fidelity model? To answer this question, we performed simulations with HF-DLD. We used a long device which contains $n_{\text{row}} = 12$ rows and $n_{\text{col}} = \lceil 4n \rceil$ columns where n is the period. For each scenario in Section 5.4 we perform simulations with the initial guess and the optimal designs in Fig. 5.5. We present the cell trajectories in Figures 5.7, 5.8, 5.9 and 5.10 for the scenarios 1, 2, 3 and 4, respectively. Here the cells in blue are softer than the cells in red. We now proceed by discussing these results briefly.

Remark. In order to test the sensitivity to the manufacturing errors, we perturb the B-spline coefficients for the optimal designs by 2% and perform the simulations again. The cell dynamics are insensitive to this amount of perturbation in all scenarios. We omit these results.

Scenario 1: $(\nu_1 = 5, \nu_2 = 10)$ **and** $\theta = 0.17$ rad **(Fig. 5.7).** Both cells zig-zag three times in the initial guess (see Fig. 5.7(a)). In the optimal design in Fig. 5.7(b) the soft cell always displaces while the stiff cell zig-zags two times. This results in a vertical

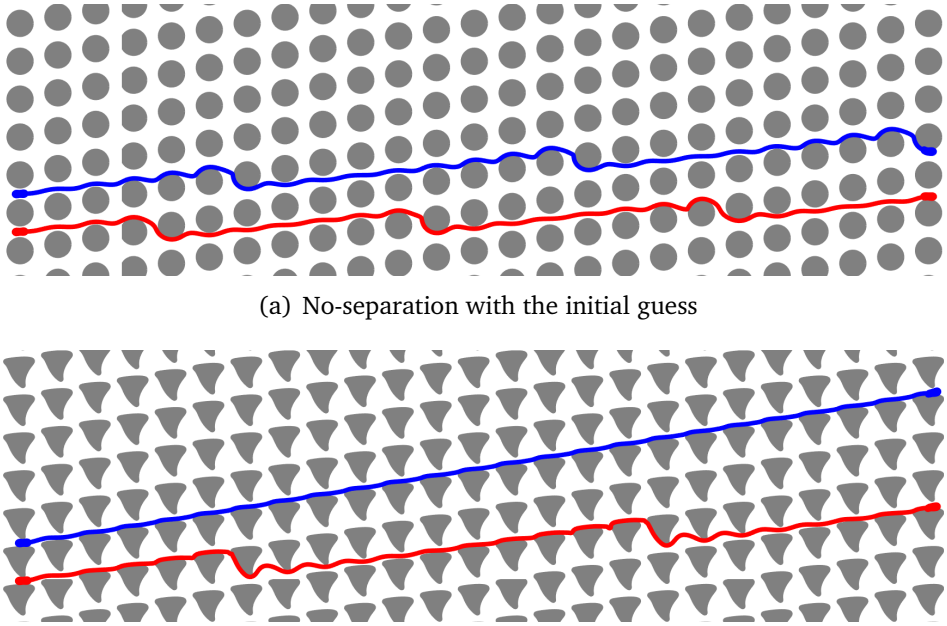


(a) No-separation with the initial guess

(b) Separation with the optimal design

Figure 5.7: *Cell trajectories in HF-DLD for scenario 1:* ($\nu_1 = 5, \nu_2 = 10$) *and* $\theta = 0.17$ rad. *The cell in blue is softer* ($\nu_1 = 5$) *than the one in red* ($\nu_2 = 10$). *The initial guess in (a) is in no-separation mode while the optimal design in (b) leads the soft cell to displace and the stiff cell to zig-zag.*

separation between the cells. The ratio of the vertical distance between the displacing cell's center and the top of the pillar underneath it at the end of the simulation to the gap is $y_f / G_y = 0.33$. There are 11 columns between the first and the second zig-zags of the stiff cell (there are fewer columns until it zig-zags for the first time since we initialize cells in the middle of the gap). Therefore, the vertical separation between the cells increases by approximately $\lambda_y = 25\mu m$ in every 11 columns. The stiff cell zig-zags less frequently in the optimal design than it does in the initial guess because the optimal design increases its vertical migration as well.



(a) No-separation with the initial guess

(b) Separation with the optimal design

Figure 5.8: *Cell trajectories in HF-DLD for scenario 2:* ($\nu_1 = 4, \nu_2 = 10$) *and* $\theta = 0.17$ rad. *The cell in blue is softer* ($\nu_1 = 4$) *than the one in red* ($\nu_2 = 10$). *The initial guess in (a) is in no-separation mode while the optimal design in (b) leads the soft cell to displace and the stiff cell to zig-zag.*

Scenario 2: ($\nu_1 = 4, \nu_2 = 10$) **and** $\theta = 0.17$ rad **(Fig. 5.8).** In this scenario, the stiff cell is the same as in the previous scenario. It zig-zags three times in the initial guess (see Fig. 5.8). The soft cell is only a little softer than the previous scenario and also zig-zags three times in the initial guess. In the optimal design in Fig. 5.8(b) the soft cell always displaces while the stiff one zig-zags two times. Here the ratio of the vertical distance between the soft cell and the pillar to the gap is $y_f/G_y = 0.37$. The stiff cell zig-zags in every ten columns, which is a little earlier than in the previous scenario. The reason is the following. The soft cell in this scenario can migrate more due to

its lower viscosity contrast. Therefore, the optimal design does not need to induce as much migration as in this scenario for the soft cell to displace. This leads the stiff cell to migrate less in the optimal design for this scenario.

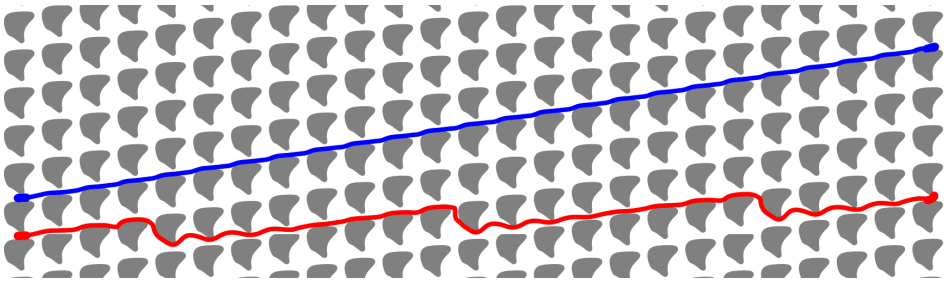
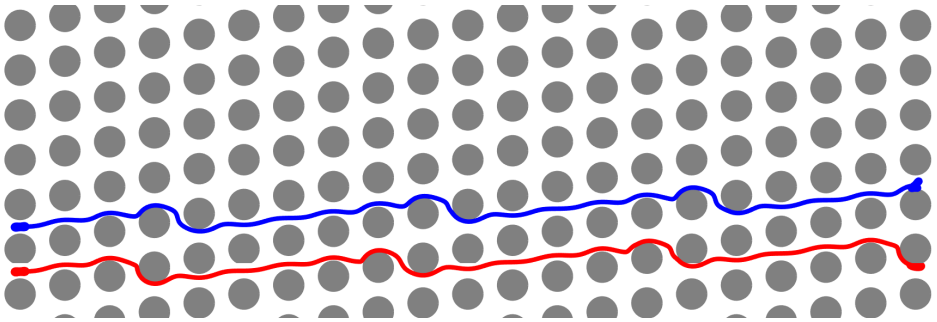
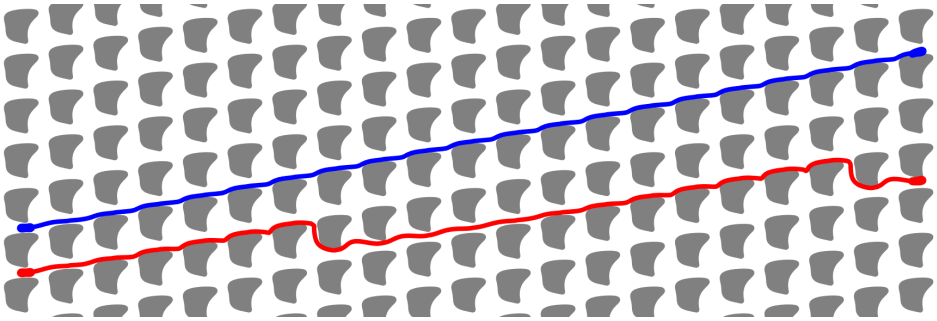


Figure 5.9: *Cell trajectories in HF-DLD with the optimal design for scenario 3:* ($\nu_1 = 5$, $\nu_2 = 50$) *and* $\theta = 0.17$ rad. *The cell in blue is softer* ($\nu_1 = 5$) *than the one in red* ($\nu_2 = 50$). *The cell trajectories in the initial guess are in Fig. 5.1(a). The initial guess is in no-separation mode while the optimal design leads the soft cell to displace and the stiff cell to zig-zag.*

Scenario 3: ($\nu_1 = 5$, $\nu_2 = 50$) **and** $\theta = 0.17$ rad **(Fig. 5.9).** In this scenario, the soft cell is the same as in the first scenario. The stiff cell has five times greater viscosity contrast value than the previous scenarios. Both cells in this scenario zig-zag three times in the initial guess (see Fig. 5.1(a)). In the optimal design in Fig. 5.9 the soft cell displaces while the stiff cell zig-zags three times. The ratio of the vertical distance between the soft cell and the pillar to the gap is $y_f/G_y = 0.36$. The stiff cell zig-zags in every seven columns, which is more frequent than the first two scenarios. Although the optimal design for this scenario needs to produce the same vertical migration as in scenario 1 because the soft cells have the same viscosity contrast values, the stiff cell is not much sensitive to this induced migration due to its high viscosity contrast. So, it zig-zags more frequently.



(a) No separation with the initial guess



(b) Separation with the optimal design

Figure 5.10: *Cell trajectories in HF-DLD for scenario 4:* ($\nu_1 = 5, \nu_2 = 10$) *and* $\theta = 0.2$ rad. *The cell in blue is softer* ($\nu_1 = 5$) *than the one in red* ($\nu_2 = 10$). *The initial guess in (a) does not sort cells while the optimal design in (b) leads the soft cell to displace and the stiff cell to zig-zag.*

Scenario 4: ($\nu_1 = 5, \nu_2 = 10$) **and** $\theta = 0.2$ rad **(Fig. 5.10).** The only difference between this scenario and the first scenario is the tilt angle of the pillars. The period becomes $n = 5$ in this scenario while it is $n = 6$ for the other scenarios. In the initial guess the soft and the stiff cells zig-zag three and four times (see Fig. 5.10(a)). In the optimal design in Fig. 5.10(b), the soft cell always displaces and the stiff cell zig-zags in every 11 columns. The ratio of the vertical distance between the soft cell and the pillar to the gap is $y_f/G_y = 0.30$. This ratio is given to be 0.37 in the LF-DLD simulation. The LF-DLD and HF-DLD simulations give similar migration of the soft cells in the optimal

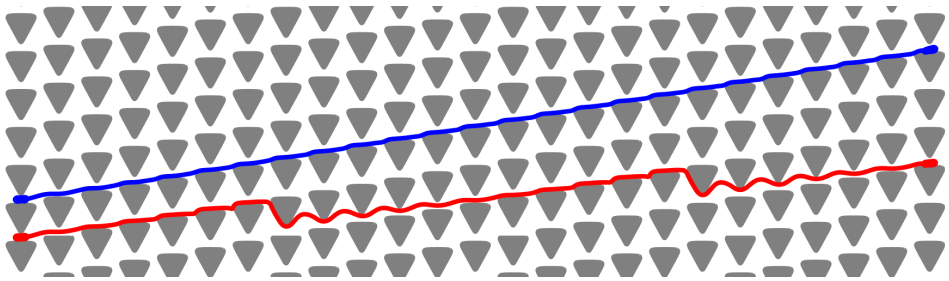
devices in the other scenarios. The LF-DLD captures the cell dynamics qualitatively in this scenario but the error in the vertical displacement is larger compared to the other scenarios. The optimal design induces so much vertical migration that the stiff cell zig-zags less frequently in it compared to the initial guess. This is also observed in scenario 1.

5.6 Simplifying Pillar Cross Section

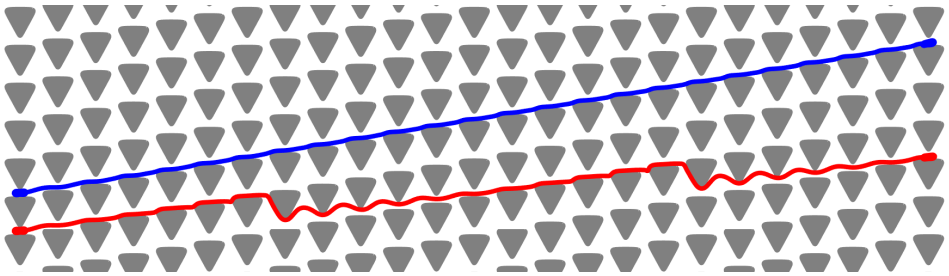
DLD arrays are manufactured by (i) drawing the pillar array pattern using a software, (ii) printing the pattern on a photo mask made of chrome quartz and (iii) manufacturing the arrays based on the mask by an etching technique (e.g., dry or wet etching of a silicon wafer [115]). Printing and etching resolutions determine manufacturing resolution. The printing is done using soft lithography, photolithography or electron beam lithography. These techniques have high resolutions and are capable of printing fine features. We are not aware of up to what precision the cross sections in the optimal designs shown in Fig. 5.5 can be manufactured. The design optimization problem we stated does not have any constraint regarding the manufacturability of the cross sections, which allows the cross sections to have sharp edges and fine features. Even if the cross sections in the optimal designs cannot be manufactured easily, they give an insight to design a device which is (i) easy to manufacture (i.e., with a basic cross section such as square, diamond, triangular without fine features), (ii) robust to uncertainties in the cells' viscosity contrast values, and (iii) robust to manufacturing errors. Based on the optimal designs we found and these criteria, we want to design a DLD device with a simpler cross sections for the cells in the first three scenarios.

We set the tilt angle of the pillar rows to 0.17 rad as in the first three scenarios. The cross sections in the optimal designs for these scenarios have a flat edge at the top and a sharp vertex at the bottom. As discussed in Section 5.5.1, these features help efficient cell sorting. That is why we suggest a triangular cross section with such configuration. We need to determine gap sizes and size of the cross section now. In the optimal designs the horizontal and the vertical gap sizes are in $G_x \in [7, 7.8]\mu m$ and $G_y \in [7, 7.2]\mu m$, respectively. This shows that cell dynamics are not as sensitive to the horizontal gap size as it is to the vertical one. Thus, we first determine the vertical gap size. We want the triangular cross section to have rounded vertices. Rounding the vertex reduces the width of the adjacent stream [107]. In order to compensate for that, we propose a vertical gap size that is slightly greater than those in the optimal designs. We set the vertical gap size to $G_y = 7.5\mu m$. To reduce the complexity of the design, We set the horizontal gap size to $G_x = 7.5\mu m$ as well. Since we enforce $\lambda_x = \lambda_y = 25\mu m$, the width and the height of the proposed cross section become $w_{\mathrm{p}} = h_{\mathrm{p}} = 17.5\mu m$. Based on these properties, we propose a triangular cross section (see Fig. 5.11(b)) of which the coordinates of the B-spline control points are tabulated in Table 5.2.

Let us assess whether the proposed design works for the first three scenarios. We present the cell trajectories in the proposed design in Fig. 5.11 for scenarios 1 and 2, and in Fig. 5.1(b) for scenario 3. In the proposed design, the cells with the viscosity contrasts $\nu = (4, 5)$ displace and those with $\nu = (10, 50)$ zig-zag. Both zig-zagging cells zig-zag in every ten columns. So the cell with $\nu = 10$ zig-zags in the proposed design with the same frequency in the optimal design and the cell with $\nu = 50$ zig-zags less frequently in the proposed design than the optimal design. We compare the objective



(a) Scenario 1: $(\nu_1, \nu_2) = (5, 10)$ and $\theta = 0.17$ rad



(b) Scenario 2: $(\nu_1, \nu_2) = (4, 10)$ and $\theta = 0.17$ rad

Figure 5.11: *Cell trajectories in HF-DLD with the proposed triangular pillar cross section for scenarios 1 and 2. See Fig. 5.1(b) for scenario 3. The cell in blue is softer than the one in red. The proposed DLD design is able to sort cells for these scenarios.*

function values for the proposed design and the optimal designs. Recall that the smaller the value is the more efficient the design is. The objective function is greater for the proposed design than the optimal designs by 22% for scenario 1, 16% for scenario 2, 30% for scenario 3. So although the optimal designs are more efficient, the proposed design has a simpler cross section and is still useful.

5.6.1 Sensitivity to Uncertainty in Viscosity Contrast Value

We investigate if the proposed design is sensitive to uncertainties in the cells' viscosity contrast values. We consider the soft and the stiff cells with $(\nu_1, \nu_2) = (5, 10)$,

respectively and perturb both ν_1 and ν_2. As mentioned before, it becomes more difficult to sort the cells when their viscosity contrast values are close. For the sensitivity analysis we increase the viscosity contrast value of the soft cell and decrease that of the stiff cell until we find a pair of viscosity contrast values for which the proposed design fails to sort. It turns out that the cells displace for the viscosity contrast $\nu \leq 8$ and zig-zag for $\nu \geq 8.5$. So the critical viscosity contrast for the separation is between 8 and 8.5 and the closest pair of the viscosity contrast values that the proposed design can separate is $(\nu_1, \nu_2) = (8, 8.5)$. In that case, the stiff cell zig-zags with the same frequency as the one with $\nu = 10$ does. Therefore, the proposed design is robust to uncertainties in the cells' viscosity contrast values.

5.6.2 Sensitivity to Manufacturing Errors

We investigate if the proposed design is sensitive to the manufacturing errors. We consider the manufacturing errors as random perturbations in the coordinates of the B-spline control points of the cross section. While doing that, we still fix $\lambda_x = \lambda_y = 25\mu m$. That is why random perturbations in the pillar cross sections result in random perturbations in gap sizes as well. We add a random noise with zero mean and standard deviations of 1%, 2%, 5%, 10% and 15% to the coordinates of the B-spline control points in Table 5.2. Then we perform simulations of the cells with $(\nu_1, \nu_2) = (5, 10)$ in the perturbed designs. The proposed design can sort these cells even for 10% perturbation, which results in 1% and 5% changes in the horizontal and vertical gap sizes, respectively. For 15% perturbation, both cells displace and hence the separation is not possible. Considering the high resolution of the micro manufacturing

techniques mentioned before, we conclude that the proposed design is robust to the manufacturing errors.

5.7 B-spline Coefficients for Optimal Designs

We parameterize pillar cross sections using uniform fifth-order B-splines with eight control points c_i, $i = 1, \cdots, 8$. In Fig. 5.5, we present the cross sections in the optimal DLD designs for four scenarios considered in Section 5.5. Additionally, we propose a triangular cross section in Section 5.6. We tabulate the coordinates of the control points for these cross sections in Table 5.2 for reproducibility of our results and manufacturing the devices.

One can reproduce a cross section in MATLAB using Algorithm 7. Here c is a matrix of size $[2 \times 8]$ and stores the coordinates of eight B-splines control points in the columns. We also tabulate the size of the cross section (h_p and w_p) in Table 5.2, so that, one can scale the produced cross section to match the correct sizes.

Algorithm 7 Reproducing cross sections from B-spline control points

// Input the coordinates of the B-spline control points, c
$c = [c(:,8); c; c(:,1); c(:,2); c(:,3)]$ `Repeat the last and the first three control points`
$[x, y] = \mathtt{spcrv}(c, 5)$ `Build a fifth-order spline curve by uniform division`
// (x, y) is a dense sequence of points on the B-spline curve representing the cross section.

Table 5.2: *Details of the optimal designs for the scenarios considered in Section 5.5 and the proposed design in Section 5.6. We tabulate the coordinates of the control points for constructing the pillar cross sections using uniform fifth-order B-splines c_i, $i = 1, \cdots, 8$; the height and the width of the cross sections h_p and w_p; the horizontal and vertical gap sizes G_x and G_y; the tilt angle of the pillar rows θ.*

	Scenario 1		Scenario 2		Scenario 3		Scenario 4		Proposed	
Control point	$x(\mu m)$	$y(\mu m)$	$x(\mu m)$	$y(\mu m)$	$x(\mu m)$	$y(\mu m)$	$x(\mu m)$	$y(\mu m)$	$x(\mu m)$	$y(\mu m)$
c_1	6.6	3.9	6.3	2.7	0.3	4.2	0.6	4.4	10.0	9.1
c_2	15.1	6.3	11.6	6.0	13.8	11.6	14.5	10.1	0	9.1
c_3	5.6	6.2	4.3	7.7	-2.0	10.6	-1.8	10.2	-10.0	9.1
c_4	-10.6	5.6	-12.8	5.1	-12.0	10.4	-12.1	7.9	-9.1	7.5
c_5	-2.0	4.1	-6.3	4.9	-6.9	-4.3	-6.9	1.4	-0.9	-7.5
c_6	-2.8	-6.5	-4.7	-6.1	-6.3	-0.9	-10.3	-6.9	0	-9.1
c_7	6.0	-13.7	5.3	-13.6	2.5	-9.2	3.6	-7.5	0.9	-7.5
c_8	1.8	-10.8	1.0	-8.6	1.0	-6.1	2.3	-9.9	9.1	7.5
$h_p(\mu m)$	18		17.9		17.9		18		17.5	
$w_p(\mu m)$	17.5		18		17.3		17.9		17.5	
$G_x(\mu m)$	7.5		7.0		7.7		7.1		7.5	
$G_y(\mu m)$	7.0		7.1		7.1		7.0		7.5	
θ (rad)	0.17		0.17		0.17		0.2		0.17	

5.8 Conclusions

We have posed designing a deterministic lateral displacement device to sort same-size red blood cells by their viscosity contrast values as a design optimization problem. Designing a device amounts to designing a pillar cross section, adjusting center-to-center distances between pillars and tilt angle of the pillar rows. We have parameterized the pillar cross section by uniform fifth-order B-splines and fix the center-to-center distance and the tilt angle. We have proposed an objective function to try to capture several factors, such as device length and sorting efficiency. We have solved the optimization problem using a stochastic optimization algorithm. The algorithm converges in $\mathcal{O}(1000)$ iterations and each iteration requires the simulation of cell flows in DLD. Our high-fidelity DLD model is computationally expensive to solve the optimization problem. In order to enable fast solution of the problem, we have proposed a low-fidelity model. We have sought optimal DLD designs for four scenarios which involve cells with similar viscosity contrast values. The optimal designs we have found can sort the cells and have had different pillar cross sections than the conventional ones (circular, triangular, square or diamond). We have compared the common features of the optimal designs with the designs proposed so far in the literature. We have also investigated the sensitivity of the optimal designs to the manufacturing errors and perturbations in the cells' viscosity contrast values. The designs have been robust to the errors and perturbations. Our study demonstrates that solving a design optimization problem systematically discovers optimal DLD designs for high resolution and efficient cell sorting. This is important since otherwise finding a design to sort cells with arbitrary deformability would require an exhaustive search.

Chapter 6

Machine Learning Acceleration of Simulations of Stokesian Suspensions

In this chapter, we propose a generic machine-learning-augmented reduced model for particulate Stokesian suspensions. The model accelerates simulations of such suspensions by replacing expensive parts of a numerical scheme with low-cost function approximations. We use multilayer perceptrons for such approximations. The model combines the perceptrons trained with high-fidelity single-particle simulations for one time step and low-fidelity simulations. Given the physical parameters of the particle, our model generalizes to arbitrary geometries and boundary conditions without the need to retrain the regression functions. It is approximately an order of magnitude faster than a state-of-the-art numerical scheme using the same number of degrees of freedom and can reproduce several features of the flow accurately. We illustrate the performance of our model on integral equation formulation of vesicle suspensions in two dimensions.

6.1 Introduction

Particulate Stokesian flows consider the motion of a collection of rigid or deformable particles (e.g., drops, capsules, cells, slender bodies, filaments, active swim-

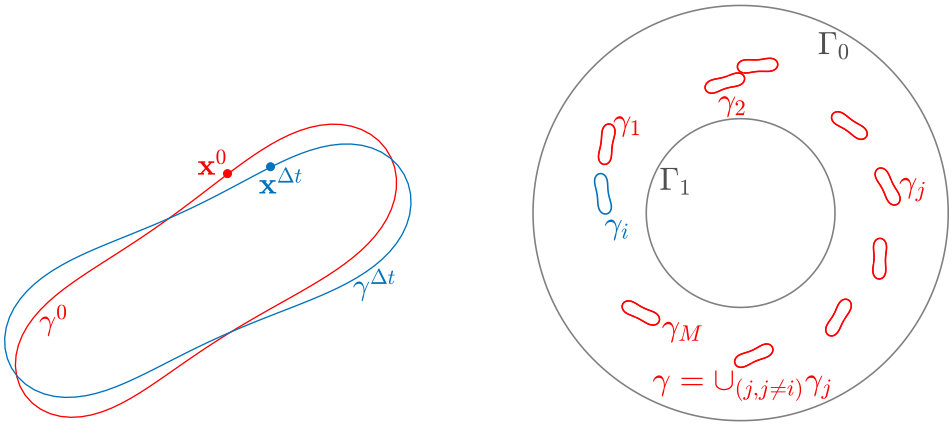
mers, possibly elastic or filled by a fluid) that are suspended in a Newtonian fluid and the particle Reynolds number is vanishingly small [16, 61, 100, 156]. Such flows find many applications in industrial processes, microfluidics, study of complex fluids, bacterial and general active flows. Due to the multiple scales, the strongly nonlinear and nonlocal coupling of the interface deformation to the background flow, and the need for long time horizons, numerical simulations, no matter what the underlying numerical method is, can be extremely expensive. In this chapter, we propose a machine-learning-augmented reduced model (MLARM) that consists of three components: (i) We use multilayer perceptrons (MLPs) to approximate several spatial nonlinear operators in our numerical scheme. We choose these operators based on their stiffness properties, computational costs, and ability to generalize to unseen data. (ii) We use high-fidelity (i.e., highly refined in space and time) simulations to train the MLPs. We run these high-fidelity simulations in "burst mode", single-particle simulations for one time step. These simulations are done only once per particle type (e.g., for various mechanical properties of a particle) and target accuracy (i.e., time step size). (iii) We combine the MLPs with low-fidelity simulations. Although we trained using single-particle, short-horizon, unconfined flows, our method enables us to conduct long-horizon simulations of suspensions with several particles in confined geometries.

The basic idea is to create a regression function that accurately captures the dynamics of the flow using the high-fidelity simulations; and then use this function in the low-fidelity numerical scheme. There exist many schemes for function approximation in high dimensions; we have opted for a multilayer perceptron [69]. We model different aspects of a flow using separate MLPs. We train a MLP on single particle

dynamics in a particle relaxation regime (no imposed flow); and we train another one for the evolution of boundary due to imposed velocity using the mode decomposition of its restriction on the boundary of the particle. Once we have trained the MLPs, the framework can be applied to any flow configuration, that is, we can vary the confined geometry, the number of particles, and the imposed boundary conditions (confined or free-space) without the need to retrain the networks. We demonstrate MLARM's capability of accurately capturing microscopic and macroscopic flow characteristics to the low-resolution simulations having the same number of degree of freedom for several benchmark problems.

A popular and effective methodology for the mathematical analysis and numerical simulation of Stokesian particulate flows is the boundary integral method (see [47, 137, 139, 148, 172] for various examples). Our overall method is based on this underlying formulation. To demonstrate our framework, we select a specific particle type, vesicle. (Although the details depend on a particle type, our model is broadly applicable to boundary integral formulations for Stokesian particulate flows). These flows are quite challenging because vesicles are deformable particles that resist bending and stretching, and are filled with a fluid. Let us note that although numerous works have used machine learning to tackle computational physics problems there has been little work on Stokesian flows. Most closely related to ours have proposed a reduced order model for unsteady quasi-one dimensional Euler flows [173]. Reduced basis functions are extracted from high-fidelity solutions and deep neural networks are used to map the flow parameters to the coefficients of the reduced basis functions. But, to our knowledge there exist no works on reduced models for Stokesian particulate

flows.



(a) A vesicle in a free-space flow

(b) Vesicles in a confined flow

Figure 6.1: *Problem setup and notation. We denote a vesicle membrane and a point on a membrane with γ and $\mathbf{x}$, respectively. (a) shows the evolution of γ^0 (red) into $\gamma^{\Delta t}$ (blue) after a time step. In (b), Γ stands for fixed boundaries (i.e., solid walls) (gray), γ_i and γ represent the i^{th} membrane (blue) and the other membranes (red).*

6.2 MLARM for Vesicle Flows

The detailed presentations of the equations governing vesicle flows and the corresponding boundary integral equation formulation are in Sections 2.2 and 2.3, respectively. MLARM is based on an alternative boundary integral formulation that allows modular MLP training. Here we present this formulation and the notation used in this chapter.

We assume that fluids in the interior and the exterior of vesicles are Newtonian with the same viscosity. Due to its simplicity, we, first, present the problem formulation for a single vesicle in a free-space flow (Fig. 6.1(a)) and then, extend it to multiple

vesicles in confined flows (Fig. 6.1(b)). Let γ denote a vesicle membrane. The points $\mathbf{x}(\alpha,t)$ on γ are given at N uniformly distributed points $\{\alpha_k = 2\pi(k-1)/N\}_{k=1}^{N}$ in the parametric domain and time t. For convenience, we introduce bending operator $\mathscr{B}$, tension operator $\mathscr{T}$, surface divergence operator $\mathscr{P}$ and stretching operator $\mathscr{M}$. These operators are defined for $\mathbf{x} \in \gamma$ as follows

$$\mathscr{B}(\mathbf{x})\mathbf{x} = -\kappa_b \mathscr{S}[\mathbf{x}_{ssss}]\mathbf{x}, \tag{6.1a}$$

$$\mathscr{T}(\mathbf{x})\sigma = \mathscr{S}[\sigma \mathbf{x}_s]\mathbf{x}, \tag{6.1b}$$

$$\mathscr{P}(\mathbf{x})\mathbf{y} = \mathbf{x}_s \cdot \mathbf{y}_s, \tag{6.1c}$$

$$\mathscr{M}(\mathbf{x}) = \mathscr{T}(\mathbf{x})\left(\mathscr{P}(\mathbf{x})\mathscr{T}(\mathbf{x})\right)^{-1}\mathscr{P}(\mathbf{x}), \tag{6.1d}$$

where $\mathscr{S}$ is the single-layer integral (2.7a), κ_b is the bending stiffness, σ is tension, and the subscript s denotes differentiation with respect to arc-length parameter. The velocity of the membrane points $\mathbf{x} \in \gamma$ in a free-space flow with the velocity field $\mathbf{u}_\infty$ is, then, given as

$$\frac{d\mathbf{x}}{dt} = (1 - \mathscr{M}(\mathbf{x}))\mathscr{B}(\mathbf{x})\mathbf{x} + (1 - \mathscr{M}(\mathbf{x}))\mathbf{u}_\infty(\mathbf{x}). \tag{6.2}$$

Here tension σ is eliminated as the stretching operator modifies the membrane velocity to enforce the inextensibility constraint (see [164] for its derivation). We rewrite (6.2) in a simple form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}_{\text{self}}[\gamma](\mathbf{x}) + V[\gamma]\mathbf{u}_\infty(\mathbf{x}), \tag{6.3}$$

where $\mathbf{u}_{\text{self}}[\gamma](\mathbf{x})$ is the velocity induced by the vesicle itself and $V[\gamma]\mathbf{u}_\infty(\mathbf{x})$ is the matrix-vector multiplication that evaluates the effect of the imposed flow $\mathbf{u}_\infty$ on the membrane velocity. We refer to matrices with upper case letters and to vectors with

lower case bold letters. Brackets following a letter mean that the preceding term depends on the solutions of integral equations involving boundaries in the brackets, e.g., $V[\gamma]$ is a matrix that requires solving integral equations for the membrane γ. We use an operator splitting method [110] for (6.3), which brings several advantages when using MLPs (they will appear later). The method divides (6.3) into two problems: advection and relaxation. Given the vesicle membrane γ^0, we, first, obtain the membrane γ^* at the intermediate step by evaluating the membrane velocity due to the imposed flow, i.e., solving

$$\frac{d\mathbf{x}}{dt} = V[\gamma^0]\mathbf{u}_\infty(\mathbf{x});\ \mathbf{x}(0) = \mathbf{x}^0 \in \gamma^0;\ t \in (0, \Delta t), \tag{6.4}$$

where Δt is the time step size. Then, the new membrane position $\mathbf{x}^{\Delta t} \in \gamma^{\Delta t}$ is given by solving the relaxation problem

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}_{\text{self}}[\gamma^*](\mathbf{x});\ \mathbf{x}(0) = \mathbf{x}^* \in \gamma^*;\ t \in (0, \Delta t). \tag{6.5}$$

The computationally expensive steps in this scheme are constructing or applying $V[\gamma]$ and solving the relaxation problem (6.5), which requires the expensive computation of $\mathbf{u}_{\text{self}}[\gamma]$. Both operators depend nonlinearly on γ. In MLARM we replace these steps with function approximations via MLPs as follows. First, for $V[\gamma]$, we need to remove the dependence on $\mathbf{u}_\infty$ in order to enable generalization to unseen $\mathbf{u}_\infty$ fields. We exploit the fact that $V[\gamma]\mathbf{u}_\infty$ depends linearly on $\mathbf{u}_\infty$. We decompose $\mathbf{u}_\infty$ using an N_f-term truncated Fourier series, $\mathbf{u}_\infty(\mathbf{x}(\alpha)) = \sum_{k=1}^{N_f} \phi_k(\alpha)\widehat{\mathbf{u}}_{\infty,k}$, where ϕ_k are Fourier basis vectors and $\widehat{\mathbf{u}}_{\infty,k}$ are the corresponding Fourier coefficients. Then, the term $V[\gamma]\mathbf{u}_\infty(\mathbf{x})$ in (6.4) becomes

$$V[\gamma]\mathbf{u}_\infty(\mathbf{x}) = \sum_{k=1}^{N_f} V[\gamma]\phi_k(\alpha)\widehat{\mathbf{u}}_{\infty,k} = \sum_{k=1}^{N_f} \Psi_k[\gamma](\alpha)\widehat{\mathbf{u}}_{\infty,k}.$$

We approximate $\{\Psi_k[\gamma]\}_{k=1}^{N_f}$ using N_f MLPs, which we term V-MLPs. Then, given an unseen γ and $\mathbf{u}_\infty$, we first compute the Ψ_k operators using the V-MLPs and then apply the inverse Fourier transform. As a bonus, $\Psi_k[\gamma]$ turns out to be linear on certain parameters such as the membrane's bending rigidity and the time step size, thus, there is no need to retrain the MLPs for these parameters. For the relaxation problem (6.5) we train another MLP that approximates the nonlinear function $\mathscr{R}(\gamma^0) = \gamma^{\Delta t}$. We call this one R-MLP. The function $\mathscr{R}$ is nonlinear in the problem parameters, therefore, retraining the MLP for different values of bending rigidity and time step size is needed. Choosing a time step size depends on viscous and bending forces on a vesicle. We train several R-MLPs for different values of time step size and bending rigidity. Given flow parameters we determine the appropriate time step size and choose the corresponding R-MLP. One can also build a parametric reduced model using the trained MLPs [8].

For multiple vesicles in confined flows Fig. 6.1(b), the velocity of the points $\mathbf{x}$ on the i^{th} membrane γ_i is given by the same equation as (6.2) with a different composition of the term $\mathbf{u}_\infty$, i.e.,

$$\frac{d\mathbf{x}}{dt} = (1-\mathscr{M}(\mathbf{x}))\mathscr{B}(\mathbf{x})\mathbf{x} + (1-\mathscr{M}(\mathbf{x}))\mathbf{u}_\infty[\gamma,\Gamma](\mathbf{x}), \tag{6.6}$$

where $\gamma = \cup_{j=1,j\neq i}\gamma_j$ and $\Gamma = \cup_{k=0}\Gamma_k$. The term $\mathbf{u}_\infty$ is velocity induced by other vesicles in the flow and the fixed boundaries on the i^{th} membrane instead of the velocity field of the background flow as in (6.2). The velocity $\mathbf{u}_\infty[\gamma,\Gamma](\mathbf{x})$ is given as

$$\mathbf{u}_\infty[\gamma,\Gamma](\mathbf{x}) = \sum_{\substack{j=1\\ j\neq i}}^{M} \int_{\gamma_j} \mathbf{G}(\mathbf{x}-\mathbf{y})\mathbf{f}(\mathbf{y})\, d\gamma_j(\mathbf{y}) + \mathscr{W}[\eta](\mathbf{x}), \tag{6.7}$$

where $\mathscr{W}[\eta]$ is the velocity induced by the fixed boundaries due to the density η on them. Computing (6.7) requires solving integral equations for the tension σ and the

density η. The inextensibility constraint (2.4) delivers the following equation for the tension

$$\mathscr{P}(\mathbf{x})\mathscr{T}(\mathbf{x})\sigma(\mathbf{x}) = -\mathscr{P}(\mathbf{x})\left(\mathscr{B}(\mathbf{x})\mathbf{x} + \mathbf{u}_\infty[\gamma,\Gamma](\mathbf{x})\right), \quad \mathbf{x}\in\gamma_i. \tag{6.8}$$

The Dirichlet velocity boundary condition on the walls provides the following equation for the density η on Γ

$$\mathbf{U}(\mathbf{x}) = -\frac{1}{2}\eta(\mathbf{x}) + \sum_{j=1}^{M}\int_{\gamma_j}\mathbf{G}(\mathbf{x}-\mathbf{y})\mathbf{f}(\mathbf{y})\,d\gamma_j(\mathbf{y}) + \mathscr{W}[\eta](\mathbf{x}), \quad \mathbf{x}\in\Gamma, \tag{6.9}$$

where $\mathscr{W}[\eta]$ is the completed double-layer integral (2.9). Boundaries do not evolve with time so the related matrices are precomputed and their application can be accelerated with fast multipole methods (FMM) [55].

The inextensibility constraint requires solving (6.8) for Lagrange multiplier (tension) for each vesicle at every time step. The solution for the i^{th} membrane can be expressed as

$$\sigma(\mathbf{x}) = \sigma_{\text{self}}[\gamma_i](\mathbf{x}) + T[\gamma_i]\bar{\mathbf{u}}_\infty[\gamma,\Gamma](\mathbf{x}), \quad \mathbf{x}\in\gamma_i, \tag{6.10}$$

where $\sigma_{\text{self}}[\gamma_i](\mathbf{x}) = -(\mathscr{P}(\mathbf{x})\mathscr{T}(\mathbf{x}))^{-1}\mathscr{P}(\mathbf{x})\mathscr{B}(\mathbf{x})\mathbf{x}$, $T[\gamma_i] = -(\mathscr{P}(\mathbf{x})\mathscr{T}(\mathbf{x}))^{-1}\mathscr{P}(\mathbf{x})$ and $\bar{\mathbf{u}}_\infty$ is the velocity due to the membranes' bending forces and the fixed boundaries. So, the first term in (6.10) is the tension due to a vesicle itself and the second one is the contribution due to the other vesicles and fixed boundaries. (6.10) resembles (6.3) for the velocity of the membrane points and we solve it using N_f+1 MLPs. That is, we use a MLP to approximate the mapping between γ_i and $\sigma_{\text{self}}[\gamma_i](\mathbf{x})$, and use N_f MLPs for the mappings between γ_i and $\{T[\gamma_i]\phi_k(s)\}_k^{N_f}$ where $\phi_k(s)$ is the Fourier basis vectors. We call the former σ-MLP and the latter T-MLPs. The input to the MLPs is the PCA coefficients of a vesicle shape γ_i. Since the same PCA basis cannot be used to represent

$\sigma_{\text{self}}[\gamma_i](\mathbf{x})$, the output to σ-MLP is the Fourier coefficients corresponding to the first 32 modes which accurately reconstruct the term.

Once the tension for all vesicles and the density on the boundaries are obtained, we use FMM to compute $\mathbf{u}_\infty[\gamma,\Gamma]$. Then we compute the term $V[\gamma_i]\mathbf{u}_\infty[\gamma,\Gamma]$ with the approximated $\Psi_k[\gamma]\widehat{\mathbf{u}}_{\infty,k}$ using N_f MLPs. Lastly, the relaxation problem (6.5) is solved. Evaluating layer potentials at points close to a boundary (fixed or vesicle) requires special quadratures [64, 96, 138, 142, 183]. These methods can be quite expensive to be implemented in MLARM. That is why, we regularize near interactions between boundaries (vesicle-vesicle, vesicle-wall) using an error function. The error due to wrong near interactions becomes important in dense vesicle suspensions and results in unphysical collisions of the boundaries. In Section 2.4.6 we have implemented a short-range repulsion force to prevent collisions. However, this also requires accurate evaluation of near interactions. So, here we employ a kinematic collision handling that is also used in simulations of emulsions [184] and red blood cells [183] in 3D. For the minimum arc-length spacing h as a threshold, whenever the distance h of a membrane point $\mathbf{x}\in\gamma_i$ to the membrane γ_j is less than h^*, the point $\mathbf{x}$ is moved in the $(\mathbf{x}_p-\mathbf{x})$ direction, where $\mathbf{x}_p$ is the projection of $\mathbf{x}$ on γ_j until $\|\mathbf{x}-\mathbf{x}_p\|=(h+h^*)/2$. Then, the membrane point on γ_j that is the closest to the projection $\mathbf{x}_p$ is moved in the same way until two membranes are separated by h^*. To have smooth perturbations in vesicle membranes, we also move the neighboring points in the same direction but for smaller amounts.

6.3 Training MLPs

In Section 6.3.1 we briefly introduce principal component analysis (PCA) that we use to represent a vesicle shape in a dimensionally reduced space. Then, in Section 6.3.2 we describe how we generate a data set for PCA and training the MLPs. In Section 6.3.3 we present the architectures of the MLPs used in MLARM. Finally, we present the pseudo-algorithm of MLARM in Section 6.3.4

6.3.1 Principal Component Analysis

We perform dimensionality reduction using principal component analysis. Here we describe how to form a reduced basis and represent a vesicle shape using this basis. A column vector $\mathbf{x} = [x_1\ x_2\ \cdots\ x_N\ y_1\ y_2\ \cdots\ y_N]^T$ of size $2N$ contains the x and y coordinates of the membrane points and represents a vesicle shape γ. Let $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_M]$ be a matrix of size $2N$-by-M that stores M vesicle shapes in the library. The singular value decomposition of $\mathbf{X}$ is

$$\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T$$

where $\mathbf{W}$ is a $2N$-by-$2N$ matrix, $\mathbf{\Sigma}$ is a $2N$-by-M rectangular diagonal matrix of the singular values of $\mathbf{X}$ and $\mathbf{V}$ is an M-by-M matrix. Let $\mathbf{C} = \mathbf{\Sigma}\mathbf{V}^T$ be a $2N$-by-M score matrix whose columns contain the representations of M samples in the orthogonally transformed space. By considering only the first N_p largest singular values and the corresponding singular vectors, we obtain $\mathbf{W}_{N_p} = [w_1\ w_2\ \cdots\ w_{N_p}]$, an $2N$-by-N_p matrix whose columns are the reduced basis vectors. Then the lower dimensional representation of a vesicle $\mathbf{x}$ is $c = \mathbf{W}_{N_p}^T\mathbf{x}$. We set $N_p = 16$ as the first 16 PCA modes capture 99% of the total energy, i.e., $(\sum_{i=1}^{16}\sigma_i^2)/(\Sigma_{i=1}^{2N}\sigma_i^2) > 99\%$.

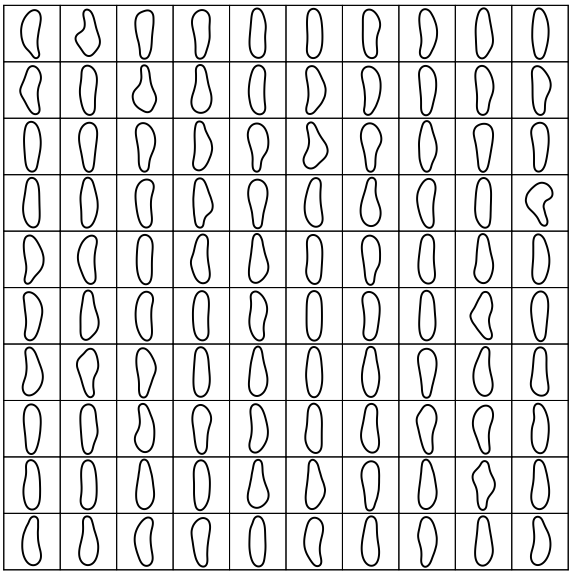


Figure 6.2: *Randomly sampled vesicles from the library. They are standardized such that their centers are at the origin, inclination angles are $\pi/2$ and arc-lengths are 1.*

6.3.2 Generating a Data Set

Training the MLPs and PCA require a library of vesicle shapes. Vesicles show wide variety of shapes in long time horizon simulations of their dense suspensions. In order to have various physical shapes in our library we perform such a simulation of vesicles having the reduced area 0.65 in a confined Taylor-Couette flow (see Fig. 2.21 for the snapshots of the simulation). We want a library that consists of distinct vesicle shapes by some metric. In order to compare the shapes obtained from the simulation, we, first, standardize the shapes so that their centers, inclination angles and arc-lengths are the same. Then, we measure the dissimilarity between the standardized shapes based on the Hausdorff metric. Our library consists of $M = 100,181$ distinct vesicle shapes. See Fig. 6.2 for 100 randomly sampled shapes from the library.

Once we have the library, we generate sets of inputs and outputs for the MLPs using a high resolution discretization, i.e., $N = 96$ points per vesicle using Algorithm 8. As the approximated terms linearly depend on translation, rotation and scaling of a vesicle, we standardize the input vesicle shapes $\mathbf{x}$ so that they have the same center and inclination angle. We also use a standard ordering of the discretization points on vesicle membranes. The first point is the one on the positive x-axis, then the other points are equally distributed along the arc-length in the counterclockwise direction. The output of R-MLP is the solution of (6.2) with $\mathbf{u}_\infty = 0$, the bending stiffness $\kappa_b = 1$ and the time step size $\Delta t = 10^{-4}$ only for one time step. This time step size is the largest one with which stable and accurate simulations in a stationary fluid can be performed using the semi-implicit time stepping scheme. For the other MLPs we obtain the outputs by solving the related integral equations for vesicles in the library. We have found that $N_f = 24$ Fourier modes accurately represent the velocity $\mathbf{u}_\infty$. Therefore, we use $N_f = 24$ MLPs to approximate the action of each V and T on the Fourier basis vectors (V-MLPs and T-MLPs). Each coefficient k has two components, imaginary and real. Additionally, since we represent the vesicles with $N = 96$ and each point has two degrees of freedom, $V\phi_k$ and $T\phi_k$ are of sizes 192 and 96. We reduce this size by subsampling $V\phi_k$ to $N = 24$ points and $T\phi_k$ to $N = 48$. The outputs to the k^{th} MLPs for these terms are vectors of size 96 that contain real and imaginary components of the terms. By choosing the subsampling rates as such we can use the same MLPs for both terms.

Algorithm 8 $[I, O_R, O_V, O_\sigma, O_T] = \texttt{generateDataSet}(\mathscr{X})$. Inputs and outputs for the MLPs

Require: Library of vesicle shapes, $\mathscr{X}$

for $\mathbf{x} \in \mathscr{X}$ **do** — For every vesicle in $\mathscr{X}$

$\mathbf{x} = \texttt{equallyDistributeInArcLength}(\mathbf{x})$ — Equally distribute points along arc-length

$[\tau, \theta, \texttt{index}] = \texttt{findStandard}(\mathbf{x})$ — Find standardization of x using Algorithm 9

$\mathbf{x}_0 = \texttt{standardize}(\mathbf{x}, \tau, \theta, \texttt{index})$ — Standardize x based on $(\tau, \theta, \texttt{index})$ using Algorithm 10

$[\mathscr{M}, \mathscr{B}, \mathscr{T}, \mathscr{P}] = \texttt{solveIEs}(\mathbf{x}_0)$ — Solve integral equations to build operators in (6.1)

$\mathbf{x}^{\Delta t} = \texttt{implicitSolveRelax}(\mathbf{x}_0, \kappa_b, \Delta t)$ — Solve relaxation ((6.2) with $\mathbf{u}_\infty = 0$) implicitly

$V = (1 - \mathscr{M})$ — Build velocity operator acting on $\mathbf{u}_\infty$

$\sigma_{\text{self}} = -(\mathscr{P}\mathscr{T})^{-1}\mathscr{P}\mathscr{B}\mathbf{x}_0$ — Find tension due to vesicle itself (6.10)

$T = -(\mathscr{P}\mathscr{T})^{-1}\mathscr{P}$ — Build tension operator acting on $\mathbf{u}_\infty$ (6.10)

$I(:, i) = \texttt{map2reducedSpace}(\mathbf{x}_0)$ — Input to MLPs is PCA coefficients of $\mathbf{x}_0$

$O_R(:, i) = \texttt{map2reducedSpace}(\mathbf{x}^{\Delta t})$ — Output to R-MLP is PCA coefficients of $\mathbf{x}^{\Delta t}$

$O_V(:, :, i) = \{V\phi_k\}_{k=1}^{N_f}$ — Output to V-MLPs is action of V on Fourier basis vectors ϕ_k

$O_\sigma(:, i) = \texttt{map2reducedSpace}(\sigma_{\text{self}})$ — Output to σ-MLP is Fourier coefficients of σ_{self}

$O_T(:, :, i) = \{T\phi_k\}_{k=1}^{N_f}$ — Output to T-MLPs is action of T on Fourier basis vectors ϕ_k

end for

Algorithm 9 $[\tau, \theta, \texttt{index}] = \texttt{findStandard}$. Translation, rotation, point order to standardize a vesicle

Require: Coordinates of membrane points, $\mathbf{x} = [x_1\ x_2\ \cdots\ x_N\ y_1\ y_2\ \cdots\ y_N]^T$

$\tau = -[\texttt{mean}(\mathbf{x}(1:N))\ \texttt{mean}(\mathbf{x}(N+1:2N))]$ — Find translation to center x at (0,0)

$\theta = \pi/2 - \texttt{getInclinationAngle}(\mathbf{x})$ — Find rotation to set inclination angle to $\pi/2$

$\texttt{index} = \texttt{findOrder}(\mathbf{x})$ — Find ordering of points

Algorithm 10 $\mathbf{x}_0 = \texttt{standardize}$, translates and rotates a vesicle and order its discretization points

Require: $\mathbf{x}, \tau, \theta, \texttt{index}$

$\mathbf{x} = \texttt{rotate}(\mathbf{x} + \tau, \theta)$ — First translate x by τ, then rotate it by θ

$\mathbf{x}_0 = \mathbf{x}(\texttt{index})$ — Finally order indices of points

6.3.3 MLP Architectures

A detailed introduction of multilayer perceptrons is beyond the scope. We only introduce definitions that are helpful to our presentation of the MLPs' architectures (see [54] for more information). A MLP has a number of parameters that are found by minimizing a loss function. Let c and $c^{\Delta t}$ be the lower dimensional representations of

vesicles $\mathbf{x}$ and $\mathbf{x}^{\Delta t}$. Then, we find the parameters of R-MLP by minimizing the mean squared error

$$\mathscr{J} = \sum_{j=1}^{M} \|\mathbf{x}_j^{\Delta t} - \widetilde{\mathbf{x}}_j^{\Delta t}\|_2, \tag{6.11}$$

where $\widetilde{\mathbf{x}}_j^{\Delta t} = W_{N_p} \widetilde{c}_j^{\Delta t}$ correspond to the MLP's approximation to the j^{th} vesicle in the data set of size M. $\|\cdot\|_2$ is the Euclidean norm. One can also measure the mean squared error based on the PCA coefficients c. We have also used a MLP trained to minimize such an error and obtained similar results. The parameters of the k^{th} V-MLP (T-MLP as well) minimize the mean square error

$$\mathscr{J} = \sum_{j=1}^{M} \|(\Psi_k)_j - (\widetilde{\Psi}_k)_j\|_2. \tag{6.12}$$

The parameters of σ-MLP minimize the mean square error between the true tension and the one reconstructed using the approximated Fourier coefficients.

All the MLPs we use consist of fully-connected layers that are defined as follows. Let W_k and b_k be the k^{th} fully connected layer's weight matrix of size n_{k+1}-by-n_k and bias vector of length n_k. Then the layer transforms an input vector z_k of length n_k into a vector $W_k z_k + b_k$ of length n_{k+1}. In the MLPs, we use batch normalization after every fully-connected layer except the last ones to shift $W_k z_k + b_k$ to zero mean and scale it to unit variance over the data set. Finally, a nonlinear function f computes an output $z_{k+1} = f(W_k z_k + b_k)$. Our choice for f is the so-called leaky rectified linear unit (leaky ReLu), i.e., $f(z) = \beta z$ if $z < 0$, $f(z) = z$ if $z \geq 0$, where β is a small constant. We set $\beta = 0.1$. The function f operates on $W_k z_k + b_k$ element-wise. A MLP consists of a cascade of such transformations. A K-layer MLP maps an input z_1 to an output z_{K+1} as

follows

$$z_2 = f(W_1 z_1 + b_1), \quad z_3 = f(W_2 z_2 + b_2), \quad \cdots, \quad z_{K+1} = f(W_K z_K + b_K).$$

The length of the vector z_{k+1} is called the width of the k^{th} layer. A MLP's parameters are the entries of the weight matrices W_k and the bias vectors b_k. We use six fully-connected layers in R-MLP. The input vector is of length 16 and the widths of the layers are 48, 96, 128, 256, 128 and 16. Therefore, the MLP has 85,920 parameters. V-MLPs and T-MLPs are the same and they consists of five fully-connected layers. The widths of the layers are 48, 96, 128, 256 and 96. The MLP has 75,632 parameters. σ-MLP has five fully-connected layers. The widths of the layers are 64, 128, 256, 128 and 64 which make up 83,584 parameters. We normalize the inputs such that they have a unit mean and zero standard deviation.

Finding parameters minimizing the loss functions requires solving a nonconvex optimization problem. We solve this problem using a stochastic gradient descent method (SGDM). SGDM computes the gradients of a loss function with respect to the MLP parameters over a randomly sampled subset of the data set which is called a minibatch. In the trainings we use minibatches of size 256. SGDM iterations are terminated either when the error stagnates or SGDM uses the entire data set 20 times to compute the gradient (i.e., after 20 epochs). SGDM has a parameter called learning rate that scales the size of a step taken in the descent direction. We set the initial learning rate to 5×10^{-4} and scaled it by a factor of 0.2 after ten epochs.

Algorithm 11 $\mathbf{x} = \texttt{destandardize}$, undoes ordering points, rotating and translating a vesicle

Require: $\mathbf{x}_0, \tau, \theta, \texttt{index}$

$\mathbf{x}(\texttt{index}) = \mathbf{x}_0$ `First re-order points`

$\mathbf{x} = \texttt{rotate}(\mathbf{x}, -\theta)$ `Rotate back by` $-\theta$

$\mathbf{x} = \mathbf{x} - \tau$ `Finally, undo translation`

Algorithm 12 `MLARM`: For a single vesicle in a free-space flow

Require: $\mathbf{u}_\infty$, time horizon T, initial vesicle configuration $\mathbf{x}$, time step size Δt

$t = 0$ `Initialize time`

while $t \leq T$ **do**

$t = t + \Delta t$

$[\texttt{area}, \texttt{length}] = \texttt{findAreaLength}(\mathbf{x})$

// First, solve advection problem (6.4)

$[\tau, \theta, \texttt{index}] = \texttt{findStandard}(\mathbf{x})$ `Find standardization`

$\mathbf{x} = \texttt{standardize}(\mathbf{x}, \tau, \theta, \texttt{index})$ `Standardize x`

$c = \mathbf{W}_{16}^T \mathbf{x}$ `Find PCA coefficients` c `of x`

$\{\Psi_k\}_{k=1}^{N_f} = \texttt{VMLP}(c)$ `Approximate` $\Psi_k = V[\gamma]\phi_k$

$\mathbf{u}_\infty = \texttt{standardize}(\mathbf{u}_\infty, 0, \theta, \texttt{index})$

$\{\widehat{\mathbf{u}}_{\infty,k}\}_{k=1}^{N_f} = \texttt{fft}(\mathbf{u}_\infty)$ `Find Fourier coefficients of` $\mathbf{u}_\infty$

$V[\gamma]\mathbf{u}_\infty = \sum_k^{N_f} \Psi_k \widehat{\mathbf{u}}_{\infty,k}$ `Reconstruct` $V[\gamma]\mathbf{u}_\infty$

$V[\gamma]\mathbf{u}_\infty = \texttt{destandardize}(V[\gamma]\mathbf{u}_\infty, 0, \theta, \texttt{index})$

$\mathbf{x} = \mathbf{x} + \Delta t V[\gamma]\mathbf{u}_\infty$ `Update x due to` $\mathbf{u}_\infty$

// Second, solve relaxation problem (6.5)

$[\tau, \theta, \texttt{index}] = \texttt{findStandard}(\mathbf{x})$

$\mathbf{x} = \texttt{standardize}(\mathbf{x}, \tau, \theta, \texttt{index})$

$c = \mathbf{W}_{16}^T \mathbf{x}_0$ `Find PCA coefficients` c `of x`

$c^{\Delta t} = \texttt{RMLP}(c)$ `Approximate PCA coefficients of` $\mathbf{x}^{\Delta t}$

$\mathbf{x}^{\Delta t} = \mathbf{W}_{16} c^{\Delta t}$ `Construct` $\mathbf{x}^{\Delta t}$ `from PCA coefficients`

$\mathbf{x}^{\Delta t} = \texttt{destandardize}(\mathbf{x}^{\Delta t}, \tau, \theta, \texttt{index})$

// Improve accuracy of $\mathbf{x}^{\Delta t}$ using correction algorithms

$\mathbf{y}^{\Delta t} = \texttt{correctShape}(\mathbf{x}^{\Delta t}, \texttt{area}, \texttt{length})$

$\mathbf{x}^{\Delta t} = \texttt{alignShape}(\mathbf{y}^{\Delta t}, \mathbf{x}^{\Delta t})$

$\mathbf{y}^{\Delta t} = \texttt{equallyDistributeInArcLength}(\mathbf{x}^{\Delta t})$

$\mathbf{x}^{\Delta t} = \texttt{alignShape}(\mathbf{y}^{\Delta t}, \mathbf{x}^{\Delta t})$

$\mathbf{x} \leftarrow \mathbf{x}^{\Delta t}$

end while

6.3.4 Pseudo-algorithm

In Algorithm 12 we present the pseudo-algorithm of MLARM only for a vesicle in a free-space flow case due to its simplicity. We use several correction algorithms in order to improve accuracy of the solution and also stability of MLARM. One of them is the area and length correction algorithm (`correctShape`) (see Section 2.4.3). After correcting the errors in a vesicle's area and arc-length, the vesicle might rotate and translate. We remove such artificial rotation and translation using `alignShape` algorithm (see Section 2.4.5). In the flows with multiple vesicles and solid walls, we compute the pairwise interactions between the boundaries using N_{LR} points to discretize a vesicle membrane. So, the pairwise interactions are computed at a low resolution.

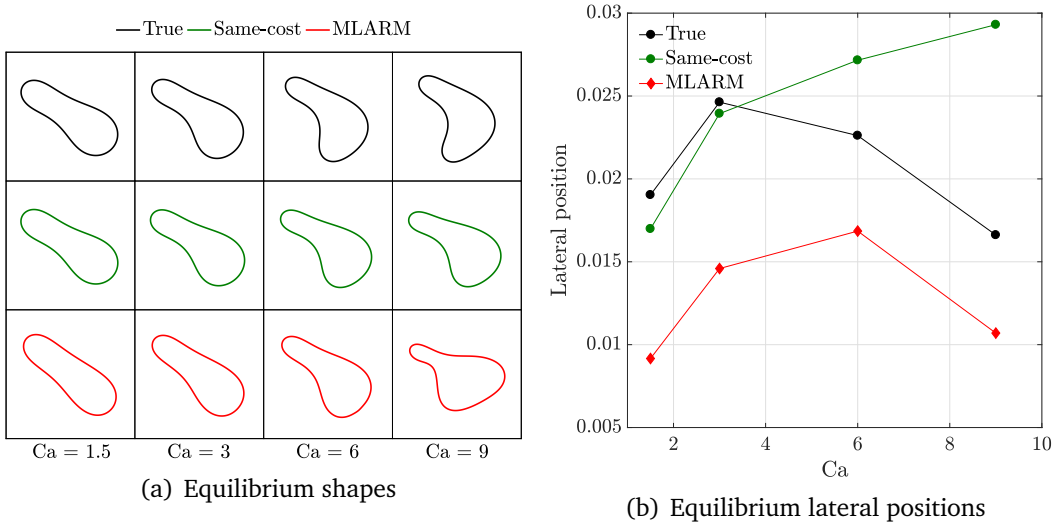
6.4 Numerical Experiments

We demonstrate MLARM's capabilities on several benchmark vesicle flows. We measure length in units of the vesicle radius r_0 (defined as the radius of a circle having the same enclosed area). The flows are characterized by capillary number $\text{Ca} = \mu r_0^3 \dot{\gamma}/\kappa_b$ where μ is the dynamic fluid viscosity and $\dot{\gamma}$ is the imposed shear rate that is varied to adjust Ca. *True* solutions are obtained using high-fidelity simulations [142]. We compare the MLARM simulations and *the same-degree-of-freedom (DOF) simulations* performed with the numerical scheme that has the same number of DOF as MLARM in terms of accuracy and computation time. We also compare with *the same-cost simulations* that have the same computation time with the MLARM simulations in terms of accuracy.

The true solutions are obtained using $N = 96$ points per vesicle and the time step size $\Delta t = 10^{-4}$. In the confined flow examples, we use 192 points to discretize the walls in the MLARM, the same-DOF, and the same-cost simulations and 256 points in the true solution. In the single vesicle examples (parabolic and shear flows, flow with curved flow lines), we use 16 PCA modes in MLARM and $N_{LR} = 16$ to solve the equations exactly at every ten time steps. In the confined flow examples, we use 32 PCA modes in order to have more degrees-of-freedom in vesicle shapes and $N_{LR} = 32$ to compute the pairwise interactions via FMM. In these examples, we do not solve the equations exactly at all. Accordingly, the number of points per vesicle is $N = 16$ in the same-DOF simulations of the single vesicle examples and $N = 32$ in those of the confined flow ones. The same-DOF simulations have $\Delta t = 10^{-4}$. We determine N and Δt for the same-cost simulations by taking several time steps with the numerical scheme for different N and Δt and matching the computation time. We set $N = 48$ and $\Delta t = 3 \times 10^{-4}$ in the same-cost simulations as they have the same CPU time as the MLARM simulations for this resolution.

6.4.1 Parabolic Flow

We consider a vesicle initialized at $y = r_0/2$ in a parabolic flow ($\mathbf{u}_\infty = r_0\dot{\gamma}\left(1-(y/W)^2\right)\mathbf{e}_x$) with $W = 10r_0$. In this flow a vesicle migrates towards the low shear rate region and reaches an equilibrium shape depending on its reduced area and Ca. We reproduce the results for a vesicle of reduced area 0.65 in [86]. The equilibrium shape is asymmetric (slipper-like) for all Ca values. Fig. 6.3(a) shows that the MLARM simulations capture the true equilibrium shapes for all Ca < 9 as accurate as



(a) Equilibrium shapes

(b) Equilibrium lateral positions

Figure 6.3: *MLARM vs. the same-cost simulations for parabolic flows. (a) Equilibrium vesicle shapes in parabolic flows with different maximum velocities. (b) We omit the results for the same-DOF simulations since they deliver equilibrium lateral positions above 0.6 for all Ca values.*

the same-cost simulations. For Ca, the scheme requires finer temporal resolution. The same-DOF simulations give erroneous equilibrium shapes (see Fig. 6.4. We present the equilibrium lateral positions of vesicles in Fig. 6.3(b) as well. While the MLARM and the same-cost simulations have similar errors in the equilibrium lateral position for $\text{Ca} = 1.5$, the MLARM simulations become more accurate for larger Ca values.

6.4.2 Cross-streamline Migration

We study cross-streamline migration of a vesicle suspended in a flow with curved flow lines. The setup is the same as in [50]. The imposed shear rate is $\dot{\gamma} = 20/r^2$ and the velocity field is $(v_\theta, v_r) = (-10/r, 0)$ where r is the distance between the vesicle's center and the origin. The value of Ca depends on r (varying from $\text{Ca} = 0.2$ at

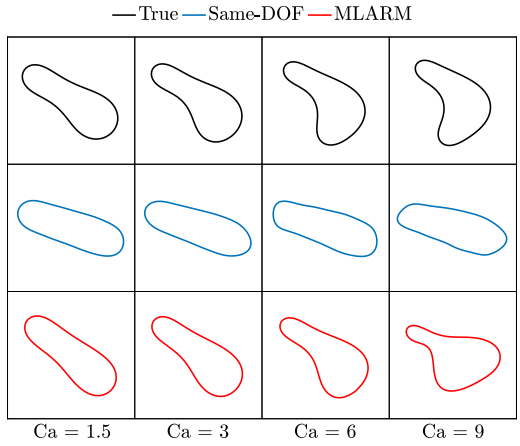
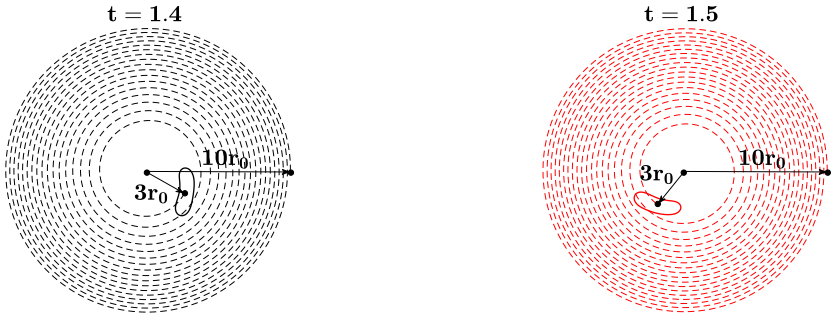


Figure 6.4: *Equilibrium vesicle shapes in a parabolic flow. The MLA simulations are more accurate than the same-DOF ones in capturing the equilibrium shapes. See the Supplemental Material for the movie of Ca = 6 case.*



(a) True simulation (b) MLA simulation

Figure 6.5: *Vesicle trajectories in a flow with curved flow lines. A vesicle initialized at* $10r_0$ *migrates towards the center in time* t*. The final radial position is* $3r_0$*. The MLARM simulation accurately captures the migration with 6% error in migration velocity.*

$r = 10r_0$ to Ca ≈ 2.2 at $r = 3r_0$). In this flow a vesicle with properties we consider (no viscosity contrast and reduced area of 0.65) migrates towards regions of high shear rates (see Fig. 6.5(a)). We measure migration velocity in the radial direction from $r = 10r_0$ to $r = 3r_0$. Figure 6.5(b) shows that the MLARM simulation captures the

migration with the error $|v_{\text{mig}}^{\text{MLARM}} - v_{\text{mig}}^{\text{True}}|/v_{\text{mig}}^{\text{True}} = 0.06$. We do not present the result of the same-cost simulation for this case since it is erroneous (error in the migration velocity is 260%). So, both low-resolution simulations cannot capture cross-streamline migration of the vesicle due to the flow line curvature.

6.4.3 Dilute Taylor-Couette Flow

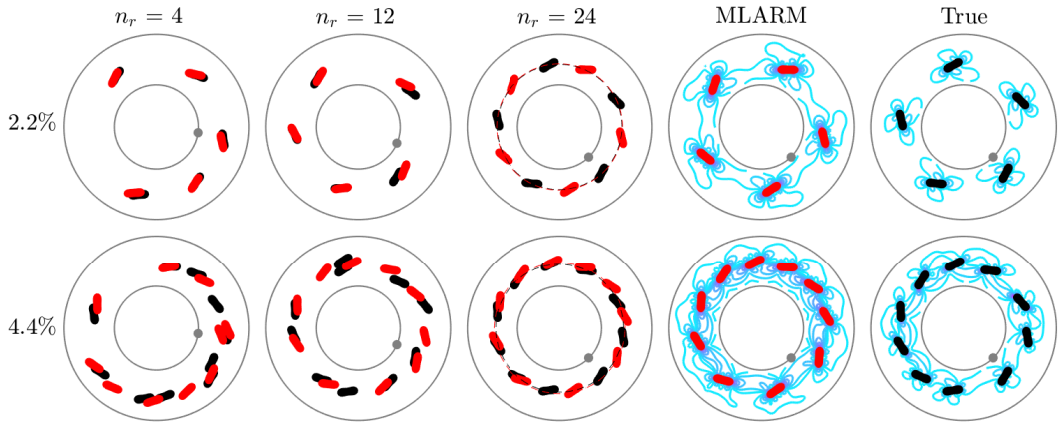
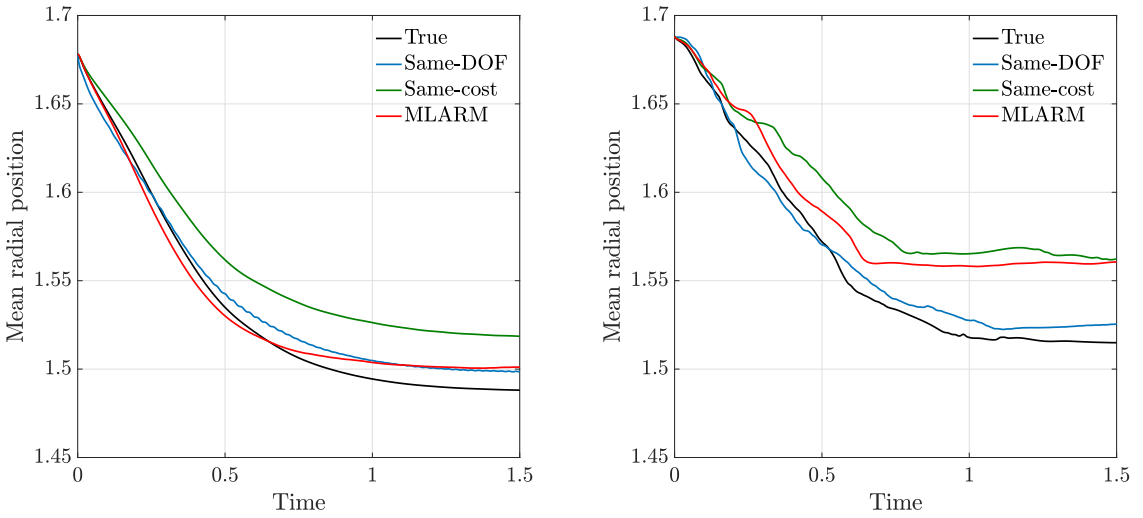


Figure 6.6: *Equilibrium organization of vesicles in their dilute suspensions in a Taylor-Couette flow (area fractions of 2.2% at the top and 4.4% at the bottom). The MLARM and true solutions are superimposed and shown in red and black, respectively. At the equilibrium of such flows, vesicles exhibit a spatial order by moving in a rim with a uniform angular interdistance. We present snapshots after n_r number of rotations of the inner circle. The dashed lines in the third column show the equilibrium rims. The last two columns demonstrate the magnitude of the perturbation in the velocity field induced by vesicles after $n_r = 24$ rotations. See the Supplemental Material for the movies.*

We consider dilute suspensions in a Taylor-Couette flow (see Fig. 6.6). Here, the inner circle rotates in the counterclockwise direction while the outer one is stationary. The distance between the inner and the outer circles is $10r_0$ and $\text{Ca} = 1.5$. In this example, the forces applied by the inner circle balance vesicles' inward migration. They, eventually, organize themselves in a rim with the same interdistance for the area

fraction of vesicles between approximately 1% and 4% [50]. We perform simulations with area fractions 2.2% and 4.4% until the inner circle completes 24 rotations. Figure 6.6 shows that closely and randomly initialized vesicles reach to approximately the same radial positions and are separated uniformly in the azimuthal direction at the equilibrium. The MLARM simulations can capture the vesicles' spatial order although the vesicles' trajectories are erroneous. The MLARM simulation becomes less accurate than the same-DOF simulation as the area fraction increases since the scheme ignores the near vesicle interactions.



(a) AF = 2.2% (b) AF = 4.4%

Figure 6.7: *Evolution of the mean radial position of the vesicles in the dilute Taylor-Couette flows.*

We present the evolution of the mean radial position of the vesicles (i.e., the mean of the distances of vesicles' centers to the inner circle's center) for area fractions 2.2% in Fig. 6.7(a) and 4.4% in Fig. 6.7(b). The figures show that the mean radial position converges as expected. The MLARM and the same-DOF simulations capture

the true mean radial position accurately in the 2.2% case. The MLARM simulation has a greater error in the 4.4% case compared to the 2.2% case. The reason is that the MLARM scheme ignores near interactions between vesicles and these interactions start dominating the dynamics as area fraction increases. For the denser case, the same-DOF simulation becomes more accurate than the MLARM simulation. In comparison with the same-cost simulation the MLARM simulation is more accurate for an area fraction 2.2%. The reason is that the same-cost simulation has a coarser temporal resolution than the MLARM simulation and capturing vesicle migration accurately requires fine temporal resolution. For an area fraction 4.4%, however, the same-cost and the MLARM simulations have similar accuracy.

6.4.4 Dense Taylor-Couette Flow

We also consider dense vesicle suspensions in Taylor-Couette flows. As in Section 2.5.6, we investigate how accurately MLARM can capture the cell-free layers formed near the walls due to vesicles' migration away from the walls. Fig. 6.8 shows frames from simulations of suspensions at area fractions of 20% at the top and 35% at the bottom. Red vesicles show the MLARM simulations and black ones are the ground truths. We plot the probability distribution of the distances of the vesicles' centers to the origin throughout the simulations in Fig. 6.9(a) for an area fraction 20% and in Fig. 6.9(b) for 35%. The MLARM and the same-DOF simulations accurately capture the cell-free layers whereas the same-cost simulation does not provide that much accuracy in the 20% area fraction case. There is no resolution that provides stable and the same computational cost as the MLARM simulation in the 35% area fraction case.

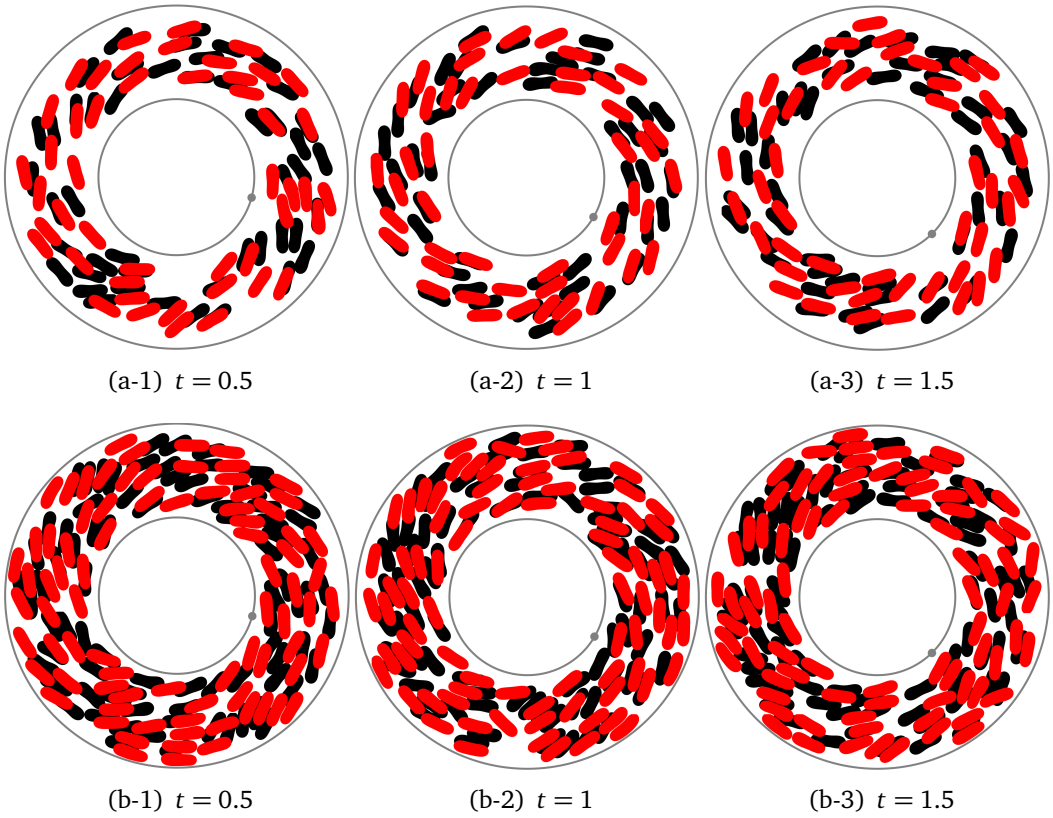
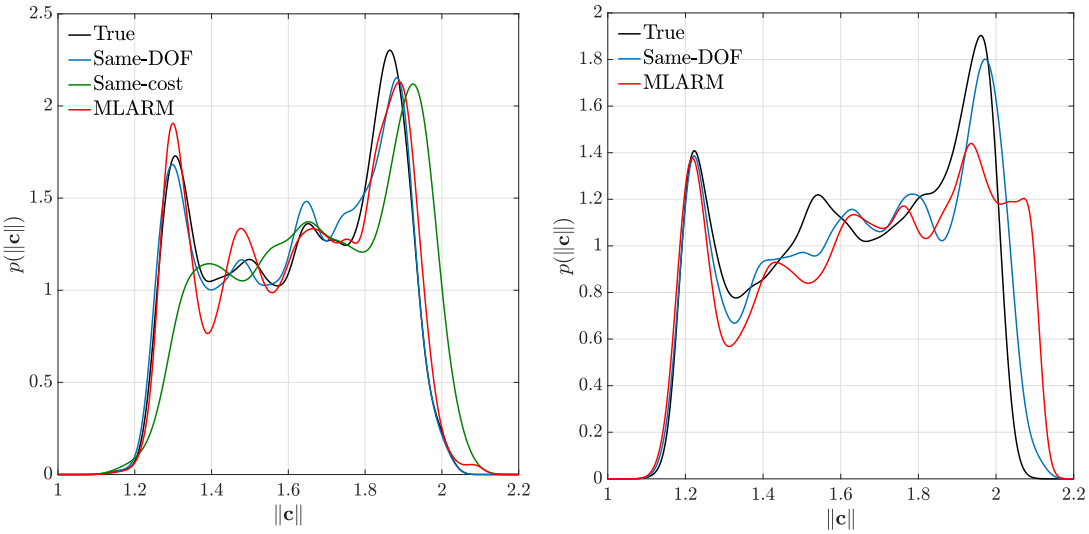


Figure 6.8: *Dense Taylor-Couette flows. The suspensions at the top and the bottom have area fractions of* 20% *and* 35%*, respectively. Vesicles in red are from MLARM simulations while those in black are the ground truths. The inner circle completes 24 revolutions in* $t = 1.5$.

Here the MLARM simulations are 10× faster than the same-DOF simulations and 25× faster than the ground truth simulations. This example clearly shows the superiority of the MLARM simulations over the low-resolution simulations.



(a) AF = 20%

(b) AF = 35%

Figure 6.9: *Statistics of vesicles in dense Taylor-Couette flows. We plot the probability distribution of the distance of the vesicle's center to the origin* $\|\mathbf{c}\|$ *throughout the simulations. Vesicles form regions near circles that are free of vesicles. The MLARM simulations accurately capture these regions.*

6.5 Conclusions

We propose a reduced model that combines multilayer perceptrons trained with high-fidelity simulations for one time step and a low-fidelity numerical scheme. In various examples of vesicle flows, we compare the model and the state-of-the-art numerical scheme. The reduced model simulations are qualitatively as accurate as and approximately an order of magnitude faster than the simulations performed with the numerical scheme using the same number of degree-of-freedom as the model. Additionally, the reduced model simulations are more accurate than the simulations performed with the numerical scheme that take the same computation time. One of the limitations of the model is that it uses a constant time step size. A reduced model

with adaptive time stepping can provide faster simulations. Additionally, we have not optimized the MLPs' architectures and training parameters. Such optimization can provide more accurate approximations. Although we presented our analysis in 2D flows (so that we can ensure highly accurate ground truth simulations), the model generalizes as is to 3D vesicle flows and other particles like deformable capsules, drops, filaments and rigid bodies.

Chapter 7

Conclusions and Future Work

7.1 Summary of the Contributions

Low-resolution simulations of vesicle suspensions. We have presented a numerical scheme that systematically integrates several correction algorithms to enable stable and accurate low-resolution simulations. Removing the need for high resolutions speeds up the simulations of vesicle suspensions significantly. We have heuristically calibrated the parameters of these algorithms to develop a black-box solver. The proposed scheme delivers statistically accurate low-resolution simulations of dense vesicle suspensions with 10× to 100× speed-up.

Mixing in vesicle suspensions. We have simulated mixing of a passive scalar in a Taylor-Couette flow with and without vesicles. We have studied effects of area fraction of vesicles, viscosity contrast between the fluids in the interior and the exterior of vesicles, initial condition of the solute, the Peclet number and mixing metric. We have found that the presence of vesicles in most cases slightly suppresses mixing because the solute can be only diffused across the vesicle interface and not advected. Additionally, we have found spatial distributions of the solute for which the default Taylor-Couette flow completely fails to mix whereas the presence of vesicles enables mixing.

Deformability-based sorting of red blood cells using a microfluidic device. Performing high-fidelity simulations we have investigated the physical mechanism that enables sorting RBCs by their deformability via a deterministic lateral displacement device. We have built a computationally efficient DLD model that captures true physics accurately. We have investigated the effects of the interior fluid viscosity and membrane elasticity of a cell on its behavior. We have found that the underlying mechanism is cell migration due to the cell's positive inclination and curved flow lines. We have also found that the technique fails for dense suspensions.

Optimal device design for efficient cell sorting. We have posed an optimization problem to systematically discover new DLD designs for efficient sorting of cells with arbitrary mechanical properties by their deformability. For fast solution of the optimization problem we have developed a low-fidelity DLD model. By considering several scenarios we have found novel DLD designs.

Machine-learning-augmented reduced model. We have presented a reduced model for Stokesian particulate flows, in particular, vesicles. This model accelerates simulations by replacing expensive parts of a numerical scheme with low-cost function approximations. We use separate multilayer perceptrons to model different aspects of the flows. Although we train the perceptrons using high-fidelity single-particle simulations for one time step and unconfined flows, our method enables us to conduct long-horizon simulations of suspensions with several particles in confined geometries. The model is predictive and faster than the low-resolution scheme.

7.2 Future Work

Main limitation of our studies is that they are in 2D. The correction algorithms we have presented for low-resolution simulations can be extended to 3D (see [111]). However, these simulations are still computationally expensive to investigate real physics problems such as mixing in vesicle suspensions and cell sorting via a microfluidic devices in 3D, let alone design optimization. We foresee that our machine-learning-augmented reduced model can significantly speed up 3D simulations of Stokesian particulate flows. That is why, its extension to 3D can be a remarkable contribution to the literature.

As mentioned, using dense suspensions in DLD devices would remove the need to dilute cell suspensions and provide high-throughput sorting. However, the DLD technique fails to sort cells in their dense suspensions. That is why, a more challenging design optimization problem is to discover optimal DLD designs that can sort cells from their dense suspensions. Our low-fidelity DLD model does not allow simulations of dense suspensions. We plan to use our machine-learning-augmented reduced model for such simulations and solving the optimization problem.

Bibliography

[1] D. Abreu, M. Levant, V. Steinberg, and U. Seifert. Fluid vesicles in flow. *Advances in Colloid and Interface Science*, 208:129–141, 2014. (page 2).

[2] A. Adrover, M. Giona, F.J. Muzzio, S. Cerbelli, and M.M. Alvarez. Analytic expression for the short-time rate of growth of the intermaterial contact perimeter in two-dimensional chaotic flows and hamiltonian systems. *Physical Review Letters*, E 58:447–458, 1998. (page 95).

[3] M. Al-Fandi, M. Al-Rousan, M. A. Jaradat, and L. Al-Ebbini. New design for the separation of microorganisms using microfluidic deterministic lateral displacement. *Robotics and Computer-Integrated Manufacturing*, 27:237–244, 2011. (page 188).

[4] S. Aland, S. Egerer, J. Lowengrub, and A. Voigt. Diffuse interface models of locally inextensible vesicles in a viscous flow. *Journal of Computational Physics*, 277:32–47, 2014. (page 19).

[5] B. K. Alpert. Hybrid Gauss-trapezoidal quadrature rules. *SIAM Journal on Scientific Computing*, 20:1551–1584, 1999. (page 27).

[6] M.M. Alvarez, F.J. Muzzio, S. Cerbelli, A. Adrover, and M. Giona. Self-similar spatiotemporal structure of intermaterial boundaries in chaotic flows. *Physical Review Letters*, 81:3395–3398, 1998. (page 95).

[7] Othmane Aouane amd Marine Thiebaud, Abdelilah Benyoussef, Christian Wagner, and Chaouqi Misbah. Vesicle dynamics in a confined Poiseuille flow: From steady state to chaos. *Physical Review E*, 90(3):033011, 2014. (pages 69, 153, 172).

[8] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46:1803–1813, 2008. (page 229).

[9] A. M. Annaswamy and A. F. Ghoniem. Active control in combustion systems. *IEEE Control Systems*, 15:49–63, 1995. (page 93).

[10] H. Aref and S. Balachandar. Chaotic advection in Stokes flows. *Physics of Fluids*, 29:3515–3521, 1986. (page 94).

[11] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32:797–823, 1995. (page 26).

[12] P. Ashwin, M.Nicol, and N.Kirkby. Acceleration of one-dimensional mixing by discontinuous mappings. *Physica A*, 130:347–363, 2002. (page 95).

[13] J. Autebert, B. Coudert, F-C. Bidard, J-Y. Pierga, S. Descroix, L. Malaquin, and J-L. Viovy. Microfluidic: An innovative tool for efficient cell sorting. *Methods*, 57:297–307, 2012. (page 127).

[14] G. R. Baker and M. J. Shelley. On the connection between thin vortex layers and vortex sheets. *Journal of Fluid Mechanics*, 215:161–194, 1990. (page 19).

[15] G.A. Barabino. Sickle cell biomechanics. *Annu. Rev. Biomed. Eng.*, 12:345–367, 2010. (page 129).

[16] Dominique Barthés-Biesel. *Microhydrodynamics and Complex Fluids*. CRC Press, 2012. (page 224).

[17] S. Batzri and E. D. Korn. Single bilayer liposomes prepared without sonication. *Biochimica et Biophysica Acta*, 298:1015–1019, 1973. (page 3).

[18] J. Beaucourt, F. Rioual, T. Séon, T. Biben, and C. Misbah. Steady to unsteady dynamics of a vesicle in a flow. *Physical Review Letter E*, 69(1), 2004. (pages 19, 174).

[19] J. P. Beech, S. H. Holm, K. Adolfsson, and J. O. Tegenfeldt. Sorting cells by size, shape and deformability. *Lab on a Chip*, 12:1048–1051, 2012. (pages 56, 82, 83, 84, 129).

[20] A. S. Bhagat, H. Bow, H. W. Houd, S. J. Tan, J. Han, and C. T. Lim. Microfluidics for cell separation. *Med. Biol. Eng. Comput.*, 49:999–1014, 2010. (page 127).

[21] T. Biben and C. Misbah. Tumbling of vesicles under shear flow within an advected-field approach. *Physical Review E*, 67:031908, 2003. (page 19).

[22] F. Bottausci, I. Mezic, C. D. Meinhart, and C. Cardonne. Mixing in the shear superposition micromixer: three-dimensional analysis. *Philos. Trans. R. Soc. Lond. Ser. A*, 362:1001–1018, 2004. (page 94).

[23] J. P. Boyd. *Chebyshev and Fourier Spectral Methods (2nd Edition, Revised)*. Dover Publications, Mineola, NY, USA, 2013. (page 95).

[24] A.V. Buys, M.-J. Van Rooy, P. Soma, D. Van Papendorp, B. Lipinski, and E. Pretorius. Changes in red blood cell membrane struture in type 2 diabetes: A scanning electron and atomic force microscopy study. *Cardiovasc. Diabetol.*, 12:25, 2013. (page 129).

[25] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, Berlin, 1987. (page 18).

[26] S. Chien. Red cell deformability and its relevance to blood flow. *Ann. Rev. Physiol.*, 49:177–192, 1987. (page 154).

[27] A. Christlieb, W. Guo, M. Morton, and J-M. Qiu. A high order time splitting method based on integral deferred correction for semi-Lagrangian vlasov simulations. *Journal of Computational Physics*, 267:7–27, 2014. (page 97).

[28] G. Coupier, B. Kaoui, T. Podgorski, and C. Misbah. Noninertial lateral migration of vesicles in bounded Poiseuille flow. *Physics of Fluids*, 20:111702, 2008. (page 157).

[29] John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press, 1947. (page 98).

[30] D. D'Alessandro, M. Dahleh, and I. Mezic. Control of mixing: a maximum entropy approach. *IEEE Trans. Automatic Control*, 44:1852–1864, 1999. (page 94).

[31] G. Danker, P. M. Vlahovska, and C. Misbah. Vesicle in poiseuille flow. *Physical Review Letters*, 102:148102, 2009. (pages 3, 22, 157).

[32] G. D'Avino. Non-Newtonian deterministic lateral displacement separator: theory and simulations. *Rheol Acta*, 52:221–236, 2013. (page 137).

[33] J. A. Davis. *Microfluidic separation of blood components through deterministic lateral displacement*. PhD thesis, Princeton University, USA, 2008. (pages 8, 128, 141, 151).

[34] J. A. Davis, D. W. Inglis, K. J. Morton, D. A. Lawrance, L. R. Huang, S. Y. Chou, J. C. Sturm, and R. H. Austin. Deterministic hydrodynamics: Taking blood apart. *PNAS*, 103(40):14779–14784, 2006. (pages 8, 127, 128, 140, 141).

[35] R. Dimova, S. Aranda, N. Bezlyepkina, V. Nikolov, K. A. Riske, and R. Lipowsky. A practical guide to giant vesicles. probing the mem- brane nanoregime via optical microscopy. *J. Phys.: Condens. Matter*, 18:S1151, 2006. (page 2).

[36] C.R. Doering and J.L. Thiffeault. Multiscale mixing efficiencies for steady sources. *Physical Review E*, 025301, 2006. (page 95).

[37] E. C. Eckstein, D. G. Bailey, and A. H. Shapiro. Self-diffusion of particles in shear flow of suspension. *Journal of Fluid Mechanics*, 79:191, 1977. (page 21).

[38] Y. Elani, R.V. Law, and O. Ces. Vesicle-based artificial cells as chemical microreactors with spatially segregated reaction pathways. *Nature Communications*, 5:5305:1–5, 2014. (page 3).

[39] M. Falcone and R. Ferretti. Convergence analysis for a class of high-order semi-Lagrangian advection schemes. *SIAM J. Numer. Anal.*, 35:909–940, 1998. (page 98).

[40] Dmitry A. Fedosov, Matti Peltomaki, and Gerhard Gompper. Deformation and dynamics of red blood cells in flow through cylindrical microchannels. *Soft Matter*, 10:4258, 2014. (pages 13, 130).

[41] Z-G. Feng and E.E. Michaelides. Hydrodynamic force on spheres in cylindrical and prismatic enclosures. *Int. J. Multiphase Flow*, 28:479–496, 2002. (page 20).

[42] Z-G. Feng and E.E. Michaelides. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *Journal of Computational Physics*, 195:602–628, 2004. (page 20).

[43] M.D. Finn, S.M. Cox, and H.M. Byrne. Mixing measures for a two-dimensional chaotic Stokes flow. *Journal of Engineering Mathematics*, 48:129–155, 2004. (page 95).

[44] D. P. G. Foures, C. P. Caulfied, and P.J. Schmid. Optimal mixing in two-dimensional plane Poiseuille flow at finite Peclet number. *Journal of Fluid Mechanics*, 748:241–277, 2014. (page 94).

[45] J. B. Freund and H. Zhao. *A high-resolution fast boundary-integral method for multiple blood cells*, chapter 3, pages 71–111. CRC Press, 2010. (pages 3, 17, 22).

[46] Jonathan B. Freund. Leukocyte margination in a model microvessel. *Physics of Fluids*, 19(2), 2007. (page 13).

[47] Jonathan B. Freund. Numerical simulation of flowing blood cells. *Annual Review of Fluid Mechanics*, 46:1:67–95, 2014. (pages 3, 17, 225).

[48] Jonathan B. Freund and M. M. Orescanin. Cellular flow in a small blood vessel. *Journal of Fluid Mechanics*, 671:466–490, 2011. (page 22).

[49] G. Ghigliotti, T. Biben, and C. Misbah. Rheology of a dilute two-dimensional suspension of vesicles. *Journal of Fluid Mechanics*, 653:489–518, 2010. (page 180).

[50] G. Ghigliotti, A. Rahimian, G. Biros, and C. Misbah. Vesicle migration and spatial organization driven by flow line curvature. *Physical Review Letters*, 106:028101, 2011. (pages 2, 157, 180, 241, 244).

[51] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, and J.Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *Journal of Computational Physics*, 169:363–426, 2001. (page 20).

[52] H. L. Goldsmith and R. Skalak. Hemodynamics. *Annual Reviews of Fluid Mechanics*, 7:213–247, 1975. (pages 3, 75, 89, 91).

[53] H.L. Goldsmith and S.G. Mason. Axial migration of particles in Poiseuille flow. *Nature*, 4781:1095–1096, 1961. (page 157).

[54] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. (page 235).

[55] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325, 1987. (pages 195, 230).

[56] J. Guck, S. Schinkinger, B. Lincoln, F. Wottawah, S. Ebert, M. Romeyke, D. Lenz, H. M. Erickson, R. Ananthakrishnan, and D. Mitchell. Optical deformability as an inherent cell marker for testing malignant transformation and metastatic competence. *Biophysical Journal*, 88(5):3689–3698, 2005. (page 127).

[57] N. Hansen and A. Ostermeier. Completeley derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195, 2001. (pages 185, 197).

[58] N. Hansen, S.D. Muller, and P. Koumoutsakos. Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations. *Parallel Problem Solving from Nature-PPSN VIII*, pages 422–431, 2002.

[59] N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.*, 11(1):1–18, 2003.

[60] N. Hansen, S.D. Muller, and P. Koumoutsakos. Evaluating the CMA evolution strategy on multimodal test functions. *Parallel Problem Solving from Nature-PPSN VIII*, 3242:282–291, 2004. (pages 185, 197).

[61] John Happel and Howard Brenner. *Low Reynolds number hydrodynamics: with special applications to particulate media*, volume 1. Springer Science & Business Media, 2012. (page 224).

[62] D. Harmon, E. Vouga, B. Smith, R. Tamstorf, and E. Grinspun. Asynchronous contact mechanics. *ACM Transactions on Graphics (TOG)*, 26, 2009. (pages 16, 20, 47).

[63] W. Helfrich. Elastic properties of lipid bilayers - theory and possible experiments. *Z. Naturf. C*, 28:693, 1973. (page 2).

[64] J. Helsing and R. Ojala. Corner singularities for elliptic problems: integral equations, graded meshes, quadrature, and compressed inverse preconditioning. *Journal of Computational Physics*, 227:8820–8840, 2008. (page 231).

[65] E. Henry, S. H. Holm, Z. Zhang, J. P. Beech, J. O. Tegenfeldt, D. A. Fedosov, and G. Gompper. Sorting cells by their dynamical properties. *Scientific Reports*, 6: 34375, 2016. (pages 130, 135, 136, 146, 170, 184).

[66] V. Hessel, H. Lowe, and F. Schonfeld. Micromixers - A review on passive and active mixing principles. *Chemical Engineering Science*, 60:2479–2501, 2005. (page 94).

[67] S. H. Holm, J. P. Beech, M. P. Barret, and J. O. Tegenfeldt. Separation of parasites from human blood using deterministic lateral displacement. *Lab on a Chip*, 11: 1326–1332, 2011. (page 128).

[68] M.N. Holme, I.A. Fedotenko, D. Abegg, L. Babel J. Althaus, F. Favarger, R. Reiter, R. Tanasescu, A. Ziegler P.L. Zaffalon, B. MÃijller, T. Saxer, and A. Zumbuehl. Shear-stress sensitive lenticular vesicles for targeted drug delivery. *Nat. Rev. Drug. Discov.*, 7(8):536–543, 2012. (page 89).

[69] K. Hornik, M. Stinchombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989. (page 224).

[70] Thomas Y. Hou, John S. Lowengrub, and Michael J. Shelley. Removing the stiffness from interfacial flows with surface tension. *Journal of Computational Physics*, 114:312–338, 1994. (page 19).

[71] L. R. Huang, E. C. Cox, R. H. Austin, and J. C. Sturm. Continuous particle separation through deterministic lateral displacement. *Science*, 304(5673):987–990, 2004. (pages 8, 56, 81, 127, 151).

[72] D. W. Inglis, J. A. Davis, R. H. Austin, and J. C. Sturm. Critical particle size for fractionation by deterministic lateral displacement. *Lab on a Chip*, 6:655–658, 2006. (pages 8, 128, 151).

[73] D. W. Inglis, M. Lord, and R. E. Nordon. Scaling deterministic lateral displacement arrays for high throughput and dilution-free enrichment of leukocytes. *Journal of Micromechanics and Microengineering*, 21(5):054024, 2011. (pages 133, 167, 168, 182).

[74] D. J. Jeffrey and A. Acrivos. The rheological properties of suspensions of rigid particles. *AIChE Journal*, 22(3):417–432, 1976. (page 21).

[75] J.L.Thiffeault. Using multi scale norms to quantify mixing and transport. *Nonlinearity*, pages R1–R44, 2012. (pages 95, 101, 102).

[76] Gökberk Kabacaoğlu and George Biros. Optimal design of deterministic lateral displacement device for viscosity contrast based cell sorting. *Physical Review Fluids*, 3:124201, 2018. (pages 9, 183).

[77] Gökberk Kabacaoğlu and George Biros. Sorting same-size red blood cells in deep deterministic lateral displacement devices. *Journal of Fluid Mechanics*, 859:433–475, 2019. (pages 8, 125, 184, 209).

[78] Gökberk Kabacaoğlu and George Biros. Machine learning acceleration of simulations of Stokesian suspensions. arXiv, 2019. (page 9).

[79] Gökberk Kabacaoğlu, Bryan Quaife, and George Biros. Quantification of mixing in vesicle suspensions using numerical simulations in two dimensions. *Physics of Fluids*, 29(2):021901, 2017. (pages 7, 21, 80, 88).

[80] Gökberk Kabacaoğlu, Bryan Quaife, and George Biros. Low-resolution simulations vesicle suspensions in 2D. *Journal of Computational Physics*, 357:43–77, 2018. (pages 6, 12).

[81] V. Kantsler and V. Steinberg. Transition to tumbling and two regimes of tumbling motion of a vesicle in shear flow. *Physical Review Letters*, 96(3), 2006. (pages 160, 170, 174).

[82] V. Kantsler, E. Segre, and V. Steinberg. Dynamics of interacting vesicles and rheology of vesicle suspension in shear flow. *Europhys. Lett*, 82:58005, 2008. (page 21).

[83] B. Kaoui, N. Tahiri, T. Biben, H. Ez-Zahraouy, A. Benyoussef, G. Biros, and C. Misbah. Complexity of vesicle microcirculation. *Physical Review E*, page 041906, 2011. (pages 2, 13).

[84] B. Kaoui, T. Krüger, and J. Harting. How does confinement affect the dynamics of viscous vesicles and red blood cells? *Soft Matter*, 8:9246, 2012. (pages 160, 170, 171, 172, 173).

[85] Badr Kaoui. Computer simulations of drug release from a liposome into the bloodstream. *Eur. Phys. J. E*, 41:20:1–6, 2018. (page 89).

[86] Badr Kaoui, George Biros, and Chaouqi Misbah. Why Do Red Blood Cells Have Asymmetric Shapes Even in a Symmetric Flow? *Physical Review Letters*, 103 (18):188101, 2009. (page 240).

[87] Badr Kaoui, Ruben J. W. Jonk, and Jens Harting. Interplay between microdynamics and macrorheology in vesicle suspensions. *Soft Matter*, 10:4735–4742, 2014. (page 21).

[88] N.M. Karabacak, P.S. Spuhler, F. Fachin, E.J. Lim, V. Pai, E. Ozkumur, J.M. Martel, N. Kojic, K. Smith, P. Chen, J. Yang, H. Hwang, B. Morgan, J. Trautwein, T.A. Barber, S.L. Stott, S. Maheswaran, R. Kapur, D.A. Haber, and M. Toner. Microfluidic, marker-free isolation of circulating tumor cells from blood samples. *Nature Protocols*, 9(3):694–710, 2014. (page 128).

[89] Kakani Katija and John O. Dabiri. A viscosity-enhanced mechanism for biogenic ocean mixing. *Nature*, 460:624–626, 2009. (page 94).

[90] S. R. Keller and R. Skalak. Motion of a tank-treading ellipsoidal particle in a shear flow. *Journal of Fluid Mechanics*, 120:27–47, 1982. (page 119).

[91] C.T. Kelley and David E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35:508–523, 1998. (page 44).

[92] D.V. Khakhar, H. Rising, and J. M. Ottino. Analysis of chaotic mixing in two model systems. *Journal of Fluid Mechanics*, 172:419–451, 1986. (page 94).

[93] Min Jun Kim and Kenneth S. Breuer. Enhanced diffusion due to motile bacteria. *Physics of Fluids*, 16(9):L78–L81, 2004. (page 94).

[94] M. Kraus, W. Wintz, U. Seifert, and R. Lipowsky. Fluid Vesicles in Shear Flow. *Physical Review Letter*, 77(17):3685–3688, 1996. (pages 2, 3).

[95] A. G. Kravchenko and P. Moin. On the effect of numerical errors in large eddy simulations of turbulent flows. *Journal of Computational Physics*, 131:310–322, 1997. (page 18).

[96] M. C. A. Kropinski. Integral Equation Methods for Particle Simulations in Creeping Flows. *Computers and Mathematics with Applications*, 38:67–87, 1999. (page 231).

[97] T. Krüger, D. Holmes, and P. V. Coveney. Deformability-based red blood cell separation in deterministic lateral displacement devices - a simulation study. *Biomicrofluidics*, 8:054114, 2014. (pages 56, 82, 83, 136, 141, 166).

[98] T. Kulrattanarak, R.G.M. van der Sman, C.G.P.H. Schröen, and R.M. Boom. Analysis of mixed motion in deterministic ratchets via experiment and particle simulation. *Microfluid Nanofluid*, 10:843–853, 2011. (page 146).

[99] Antonio Lamura and Gerhard Gompper. Dynamics and rheology of vesicle suspensions in wall-bounded shear flow. *Europhys. Lett.*, 102:28004, 2013. (page 21).

[100] Eric Lauga and Thomas R. Powers. The hydrodynamics of swimming microorganisms. *Reports on Progress in Physics*, 72:096601, 2009. (page 224).

[101] D. Leighton and A. Acrivos. The shear-induced migration of particles in concentrated suspensions. *Journal of Fluid Mechanics*, 181:415–439, 1987. (pages 21, 80).

[102] F. Leiken, C. Coulliette, A. J. Mariano, E. H. Ryan, L.K. Shay, G. Haller, and J. Marsden. Pollution release tied to invariant manifolds: a case study for the coast of Florida. *Physica D*, 210:1–20, 2005. (page 93).

[103] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, Philadelphia, PA, USA, 2007. (page 99).

[104] Xuejin Li, Petia Vlahovska, and George Em Karniadakis. Continuum- and particle-based modeling of shapes and dynamics of red blood cells in health and disease. *Soft Matter*, 9:28–37, 2013. (pages 3, 17).

[105] Z. Lin, J.L. Thiffeault, and C.R. Doering. Optimal stirring strategies for passive scalar mixing. *Journal of Fluid Mechanics*, 675:465–476, 2011. (page 94).

[106] Zhin Lin, Jean-Luc Thiffeault, and Stephen Childress. Stirring by squirmers. *Journal of Fluid Mechanics*, 669:167–177, 2011. (page 94).

[107] K. Loutherback, K.S. Chou, J. Newman, J. Puchalla, R.H. Austin, and J.C. Sturm. Improved performance of deterministic lateral displacement arrays with triangular posts. *Microfluid Nanofluid*, 9:1143–1149, 2010. (pages 187, 208, 217).

[108] K. Loutherback, J. D'Silva, L. Liu, A. Wu, R.H Austin, and J.C. Sturm. Deterministic separation of cancer cells from blood at 10 mL/min. *AIP Advances*, 2(4): 042107, 2012. (pages 128, 187).

[109] Libin Lu, Abtin Rahimian, and Denis Zorin. Contact-aware simulations of particulate stokesian suspensions. *Journal of Computational Physics*, 347:160–182, 2017. (page 19).

[110] Shev Macnamara and Gilbert Strang. *Operator Splitting*, chapter 3, pages 97–116. Springer, 2016. (page 228).

[111] Dhairya Malhotra, Abtin Rahimian, Denis Zorin, and George Biros. A parallel algorithm for long-timescale simulation of concentrated vesicle suspensions in three dimensions. 2017. (pages 4, 10, 13, 14, 17, 19, 251).

[112] Gary Marple, Alex Barnett, Adrianna Gillman, and Shravan Veerapaneni. A fast algorithm for simulating multiphase flows through periodic geometries of arbitrary shape. *SIAM J. Sci. Comput.*, 38(5):B740–B772, 2016. (page 19).

[113] G. Mathew, I. Mezic, and L. Petzold. A multiscale measure for mixing. *Physica D*, 211:23–46, 2005. (pages 94, 95, 101).

[114] G. Mathew, I. Mezic, S. Grivopoulos, U. Vaidya, and L. Petzold. Optimal control of mixing in Stokes fluid flows. *Journal of Fluid Mechanics*, 580:261–281, 2007. (page 95).

[115] J. McGrath, M. Jimenez, and H. Bridle. Deterministic lateral displacement for particle separation: a review. *Lab on a Chip*, 14:4139, 2014. (pages 131, 132, 135, 153, 196, 204, 216).

[116] V.V. Meleshko and H. Aref. A blinking rotlet model for chaotic advection. *Physics of Fluids*, 8:3215–3217, 1996. (page 94).

[117] S. Meßlinger, B. Schmidt, H. Noguchi, and G. Gompper. Dynamical regimes and hydrodynamic lift of viscous vesicles under shear. *Physical Review E*, 80:011901, 2009. (pages 130, 157, 170).

[118] C. Misbah. Vacillating breathing and tumbling of vesicles under shear flow. *Physical Review Letters*, 96(2), 2006. (pages 2, 170).

[119] C. Misbah. Vesicles, capsules and red blood cells under flow. *Journal of Physics: Conference Series*, 392:012005, 2012. (pages 3, 130, 131).

[120] N. Mohandas and P.G. Gallagher. Red cell membrane: past, present and future. *Blood*, 112(10):3939–3948, 2008. (pages 154, 160, 202).

[121] Kahtrin Müller, Dmitry A. Fedosov, and Gerhard Gompper. Margination of micro- and nano-particles in blood flow and its effect on drug deliver. *Scientific Reports*, 4:4871, 2014. (page 21).

[122] F.J. Muzzio and P.D. Swanson. The statistics of stretching and stirring in chaotic flows. *Physics of Fluids*, A 3:822–834, 1991. (page 94).

[123] Vivek Narsimhan, Hong Zhao, and Eric S. G. Shaqfeh. Coarse-grained theory to predict the concentration distribution of red blood cells in wall-bounded couette flow at zero reynolds number. *Physics of Fluids*, 25:061901, 2013. (pages 21, 69).

[124] H. Noguchi and D. G. Gompper. Shape transitions of fluid vesicles and red blood cells in capillary flows. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 102:14159–14164, 2005. (pages 2, 3, 131).

[125] P. Olla. The lift on a tank-treading ellipsoidal cell in a shear flow. *Journal de Physique II, EDP Sciences*, 7(10):1533–1540, 1997. (pages 157, 170).

[126] S. A. Orszag. Numerical simulation of incompressible flows within simple boundaries: Accuracy. *Journal of Fluid Mechanics*, 49:75–112, 1971. (page 18).

[127] S. A. Orszag. Fourier series on spheres. *Monthly Weather Review*, 102:56–75, 1974. (page 18).

[128] J. M. Ottino. Mixing, chaotic advection, and turbulence. *Annu. Rev. Fluid Mech.*, 22:207–253, 1990. (page 94).

[129] J.M. Ottino and S. Wiggins. Introduction: mixing in microfluidics. *Philos. Trans. R. Soc. Lond. Ser. A*, 362:923–935, 2004. (page 94).

[130] T-W. Pan, L. Shi, and R. Glowinski. A DLM/FD/IB method for simulating cell/cell and cell/particle interaction in microchannels. *Chinese Annals of Mathematics, Series B*, 31B-6:975–990, 2010. (page 20).

[131] G. S. Patterson and S. A. Orszag. Spectral calculations of isotropic performance of a subgrid scale model can be improved by turbulence: Efficient removal of aliasing interactions. *Physics of Fluids A*, 14:2538–2541, 1974. (page 18).

[132] Thomas Podgorski, Natacha Callens, Christophe Minetti, Gwennou Coupier, Frank Dubois, and Chaouqi Misbah. Dynamics of vesicle suspensions in shear flow between walls. *Microgravity Sci. Technol.*, 23:263–270, 2011. (pages 21, 75, 130).

[133] A. S. Popel and P. C. Johnson. Microcirculation and Hemorheology. *Annu. Rev. Fluid Mech.*, 37:43–69, 2005. (pages 89, 128, 153).

[134] H. Power. The completed double layer boundary integral equation method for two-dimensional Stokes flow. *IMA Journal of Applied Mathematics*, 51(2): 123–145, 1993. (page 17).

[135] H. Power and G. Miranda. Second kind integral equation formulation of stokes' flows past a particle of arbitrary shape. *SIAM Journal of Applied Mathematics*, 47(4):689–698, 1987. (page 17).

[136] C. Pozrikidis. The Axisymmetric Deformation Of A Red Blood Cell In Uniaxial Straining Stokes Flow. *Journal of Fluid Mechanics*, 216:231–254, 1990. (page 2).

[137] C. Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press, New York, NY, USA, 1992. (pages 194, 225).

[138] C. Pozrikidis. A Spectral-Element Method for Particulate Stokes Flow. *Journal of Computational Physics*, 156:360–381, 1999. (page 231).

[139] C. Pozrikidis. Interfacial dynamics for Stokes flow. *Journal of Computational Physics*, 169:250–301, 2001. (pages 17, 225).

[140] C. Pozrikidis. Numerical simulation of the flow-induced deformation of red blood cells. *Annals of Biomedical Engineering*, 31(10):1194–1205, 2003. (pages 3, 17).

[141] Costas Pozrikidis. Interfacial dynamics for Stokes flow. *Journal of Computational Physics*, 169:250–301, 2001. (pages 3, 17).

[142] Bryan Quaife and George Biros. High-volume fraction simulations of two-dimensional vesicle suspensions. *Journal of Computational Physics*, 274:245–267, 2014. (pages 4, 7, 12, 14, 16, 17, 26, 27, 36, 38, 180, 231, 239).

[143] Bryan Quaife and George Biros. High-order adaptive time stepping for vesicle suspensions with viscosity contrast. *Procedia IUTAM*, 16:89–98, 2015. (pages 4, 14).

[144] Bryan Quaife and George Biros. Adaptive time stepping for vesicle suspensions. *Journal of Computational Physics*, 306:478–499, 2016. (pages 6, 12, 15, 18, 28, 35).

[145] R. Quek, D. V. Le, and K.-H Chiam. Separation of deformable particles in deterministic lateral displacement devices. *Physical Review E*, 83:056301, 2011. (pages 13, 82, 83, 136, 141, 146, 166).

[146] Abtin Rahimian, Shravan K. Veerapaneni, and George Biros. Dynamic simulation of locally inextensible vesicles suspended in an arbitrary two-dimensional domain, a boundary integral method. *Journal of Computational Physics*, 229: 6466–6484, 2010. (pages 4, 17, 21, 51, 53).

[147] Abtin Rahimian, Shravan K. Veerapaneni, Denis Zorin, and George Biros. Boundary integral method for the flow of vesicles with viscosity contrast in three dimensions. *Journal of Computational Physics*, 298:766–786, 2015. (pages 13, 17, 18, 19, 43).

[148] J.M. Rallison and A. Acrivos. A numerical study of the deformation and burst of a viscous drop in an extensional flow. *Journal of Fluid Mechanics*, 89:191–200, 1978. (pages 17, 225).

[149] S. Ranjan, K. K. Zeming, R. Jureen, D. Fisher, and Y. Zhang. DLD pillar shape design for efficient separation of spherical and non-spherical bioparticles. *Lab on a Chip*, 14:4250, 2014. (pages 135, 188).

[150] A. Robert. A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean*, 19:35–46, 1981. (page 95).

[151] D. Rossinelli, Y-H. Tang, K. Lykov, D. Alexeev, M. Bernaschi, P. Hadjidoukas, M. Bisson, W. Joubert, C. Conti, G. Karniadakis, M. Fatica, I. Pivkin, and P. Koumoutsakas. The In-Silico Lab-on-a-Chip: Petascale and High-Throughput Simulations of Microfluidics at Cell Resolution. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis: SC '15, Austin, Texas*, pages 2:1–2:12, 2015. (pages 134, 141, 187).

[152] D. Rothstein, E. Henry, and J.P. Gollub. Persistent patterns in transient chaotic fluid mixing. *Nature*, 401:770–772, 1999. (page 95).

[153] E. Sackmann. Supported membranes: Scientific and practical applications. *Science*, 271:43–48, 1996. (page 2).

[154] U. Seifert. Configurations of fluid membranes and vesicles. *Advances in Physics*, 46:13–137, 1997. (page 2).

[155] U. Seifert. Fluid membranes in hydrodynamic flow fields: Formalism and an application to fluctuating quasispherical vesicles in shear flow. *Eur. Phys. J. B*, 8:405–415, 1999. (page 2).

[156] Michael J. Shelley and Jun Zhang. Flapping and Bending Bodies Interacting with Fluid Flows. *Annual Review of Fluid Mechanics*, 43:449–465, 2011. (page 224).

[157] J. Siepmann and F. Siepmann. Modeling of diffusion controlled drug delivery. *J. Control Release*, 161(2):351–362, 2012. (page 89).

[158] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5(3):506–517, 1968. (page 95).

[159] S. Suresh, J. Spatz, J. P. Mills, A. Micoulet, M. Dao, C. T. Lim, M. Beil, and T. Seufferlein. Connections between single-cell biomechanics and human disease states: gastrointestinal cancer and malaria. *Acta Biomaterialia*, 1(1):15–30, 2005. (pages 127, 153).

[160] G. Tomaiuolo. Biomechanical properties of red blood cells in health and disease towards microfluidics. *Biomicrofluidics*, 8:051501, 2014. (page 153).

[161] V.P. Torchilin. Recent advances with liposomes as pharmaceutical carriers. *Nat. Rev. Drug. Discov.*, 4(2):145–160, 2005. (page 89).

[162] N. L. Trefethen. *Spectral Methods in MATLAB*. SIAM, Philadelphia, PA, USA, 2000. (page 95).

[163] Patrick T. Underhill, Juan P. Hernandez-Ortiz, and Michael D. Graham. Diffusion and spatial correlations in suspensions of swimming particles. *Physical Review Letters*, 100:248101, 2008. (page 94).

[164] S. K. Veerapaneni, D. Gueyffier, D. Zorin, and G. Biros. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D. *Journal of Computational Physics*, 228(7):2334–2353, 2009. (pages 4, 17, 24, 227).

[165] S. K. Veerapaneni, A. Rahimian, G. Biros, and D. Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *Journal of Computational Physics*, 230:5610–5634, 2011. (pages 4, 6, 15, 17, 43, 44).

[166] R. Vernekar and T. Krüger. Breakdown of deterministic lateral displacement efficiency for non-dilute suspensions: A numerical study. *Medical Engineering and Phsyics*, 37:845–854, 2015. (pages 82, 83, 133, 136, 137, 141, 166, 168, 182).

[167] R. Vernekar, T. Krüger, K. Loutherback, K. Morton, and D. Inglis. Anisotropic permeability in deterministic lateral displacement arrays. *Lab on a Chip*, 17: 3318–3330, 2017. (page 146).

[168] P. M. Vlahovska and R.S. Gracia. Dynamics of a viscous vesicle in linear flows. *Physical Review E*, 75:016313, 2007. (pages 154, 157, 170).

[169] Etienne Vouga, David Harmon, Rasmus Tamstorf, and Eitan Grinspun. Asynchronous variational contact mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200:2181–2194, 2011. (pages 16, 20, 47).

[170] B. A. Wade, A. Q. M. Khaliq, M. Yousuf, J. Vigo-Aguiar, and R. Deininger. On the smoothing of the Crank-Nicolson scheme and higher order schemes for pricing barrier options. *J. Comp. App. Math.*, 204:144–158, 2007. (page 99).

[171] C.H. Wang and A. S. Popel. Effect of red blood cell shape on oxygen transport in capillaries. *Mathematical Biosciences*, 116:89–110, 1993. (page 89).

[172] Jingtao Wang, Jinxia Liu, Junjie Han, and Jing Guan. Effects of Complex Internal Structures on Rheology of Multiple Emulsions Particles in 2D from a Boundary Integral Method. *Phys. Rev. Lett.*, 110:066001, 2013. (page 225).

[173] Q. Wang, J.S. Hesthaven, and D. Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of Computational Physics*, in press, 2019. (page 225).

[174] D. Xiu and G.E. Karniadakis. A semi-Lagrangian high-order method for the Navier-Stokes equations. *Journal of Computational Physics*, 172:658–684, 2001. (pages 95, 98).

[175] S. Ye, X. Shao, Z. Yu, and W. Yu. Effects of the particle deformability on the critical separation diameter in the deterministic lateral displacement device. *Journal of Fluid Mechanics*, 743:60–74, 2014. (pages 13, 82, 83, 136, 141).

[176] G. K. Youngren and A. Acrivos. Stokes flow past a particle of arbitrary shape: a numerical method of solution. *Journal of Fluid Mechanics*, 69:377–403, 1975. (page 17).

[177] K. K. Zeming, S. Ranjan, and Y. Zhang. Rotational separation of non-spherical bioparticles using I-shaped pillar arrays in a microfluidic device. *Nature Communications*, 4:1625, 2013. (pages 128, 135, 188).

[178] K.K. Zeming, T. Salafi, C-H. Chen, and Y. Zhang. Asymmetrical Deterministic Lateral Displacement Gaps for Dual Functions of Enhanced Separation and Throughput of Red Blood Cells. *Sci. Rep.*, 6:22934, 2016. (pages 188, 210).

[179] Z. Zhang, E. Henry, G. Gompper, and D. A. Fedosov. Behavior of rigid and deformable particles in deterministic lateral displacement devices with different post shapes. *The Journal of Chemical Physics*, 143:243145, 2015. (pages 13, 82, 83, 136, 141, 151, 157, 184, 188).

[180] Hong Zhao and Eric S. G. Shaqfeh. The dynamics of a vesicle in simple shear flow. *Journal of Fluid Mechanics*, 674:578–604, 2011. (page 22).

[181] Hong Zhao and Eric S. G. Shaqfeh. The shape stability of a lipid vesicle in a uniaxial extensional flow. *Journal of Fluid Mechanics*, 719:345–361, 2013.

[182] Hong Zhao and Eric S. G. Shaqfeh. The dynamics of a non-dilute vesicle suspension in simple shear flow. *Journal of Fluid Mechanics*, 725:709–731, 2013. (pages 3, 17, 22).

[183] Hong Zhao, Amir H.G. Isfahani, Luke N. Olson, and Jonathan B. Freund. A spectral boundary integral method for flowing blood cells. *Journal of Computational Physics*, 229:3726–3744, 2010. (page 231).

[184] A. Z. Zinchenko and R. H. Davis. Algorithm for direct numerical simulation of emulsion flow through a granular material. *Journal of Computational Physics*, 227:7841–7888, 2008. (page 231).

[185] R. Zvan, K. R. Vetzal, and P. A. Forsyth. PDE methods for pricing barrier options. *J. Econ. Dynamics & Control*, 24:1563–1590, 2000. (page 99).